Advanced Signal Processing Techniques for Underwater Acoustic Communication Networks

Chunshan Liu

Doctor of Philosophy (Ph.D.)

Department of Electronics

University of York

November 2011

Abstract

In this thesis, we develop and investigate novel signal processing techniques for underwater acoustic communication networks. Underwater acoustic channels differ from radio communication channels in the lower speed of signal propagation, richer and often sparse multipath arrivals, and more severe Doppler effect. Therefore, many signal processing techniques developed for radio communications may not work equivalently well for underwater acoustic channels.

To investigate signal processing techniques in underwater acoustics, efficient simulation of signal transmission is required. Specifically, there is requirement for accurate simulation of doubly-selective underwater channels for different acoustic environments. In this thesis, a low-complexity channel simulator has been developed for scenarios with moving transmitter/receiver. The simulator is based on efficient generation of time-varying channel impulse response obtained using interpolation over a set of waymark impulse responses for a relatively small number of sampling points on the transmitter/receiver trajectory. The waymark impulse responses are generated using an acoustic field computation method, which is the most computationally expensive part of the simulator. To reduce the trajectory sampling rate, and thus, to reduce the complexity of the field computation, an approach for adjusting the time-varying multipath delays has been developed. For setting the trajectory sampling interval, a simple rule has been proposed, based on the waveguide invariant theory. To further reduce the simulator complexity, local spline interpolation is exploited. The developed simulator has been verified by comparing the simulated data with data from real ocean experiments. In particular, applying simulated data to an OFDM modem shows similar performance with that obtained from the data of a deep water experiment.

In communication networks, knowledge of positions of communication nodes is important for improving the system performance. A multi-source localization technique has been proposed based on the matched field (MF) processing. The technique locates the nodes by solving a set of basis pursuit de-noising (BPDN) problems corresponding to a set of source frequencies. An efficient technique combining the homotopy approach and coordinate descent search has been developed to solve the BPDN problem. Further reduction in the complexity has been achieved by applying a position grid refinement method. Verified using simulated data generated by the proposed simulator and data from real experiment, the proposed technique outperforms other MF techniques in resolving sources positioned closely to each other, tolerance to noise and capability of locating multiple sources.

To provide reliable localization based on MF techniques, accurate knowledge of the underwater acoustic environment is essential. However, such a knowledge is not always available. Estimating uncertain environmental parameters can be achieved using MF inversion techniques. This requires solving a global optimization problem. Several global optimization algorithms have been investigated and an algorithm combining the simulated annealing and downhill simplex method has been applied for estimating the sound speed profile in a deep water scenario. Accurate MF localization results have been demonstrated when using the estimated sound speed profile.

A very important task of communication receivers is accurate channel estimation. The knowledge of node positions and the environment can be exploited for enhancing the channel estimation accuracy and reducing the estimation complexity. This knowledge can be used to define the structure of the channel impulse response, such as the multipath spread and the sparsity. A channel estimator exploiting the channel sparsity estimated from the node positions has been proposed and investigated. The sparse taps of the channel impulse response are identified by solving a BPDN problem. The estimator employs an iterative tap-by-tap processing and uses local splines to interpolate the time-varying tap coefficients. This allows reduction in the complexity and memory requirement, whereas providing a high estimation accuracy.

Contents

List of Figures

List of Tables

Acknowledgements

First and foremost I offer my sincerest gratitude to my supervisor, Dr. Yuriy Zakharov, who has supported me throughout my Ph.D. research with his patience and knowledge. He has been always available to advise me and encourage me. I am deeply impressed by his preciseness, patience, enthusiasm and immense knowledge. I could not have imagined having a better advisor and mentor for my Ph.D study.

Besides my supervisor, I would like to thank Teyan Chen, Peng Li and many other colleagues in Communication Research Group for the stimulating discussions, useful suggestions, help on my research and for all the fun we have had in the past three years.

This thesis is dedicated to my parents, Shutian Liu and Tianling Zuo, who give birth to me and support me with endless encouragement and patience. This thesis is also dedicated to my wife, Rui Zhang, whose love and continued support enabled me to overcome the frustrations in my study.

Finally, I would dedicate this thesis to my grandmother.

Declaration

Some of the research presented in this thesis has resulted in some publications. These publications are listed at the end of Chapter 1.

All work presented in this thesis as original is so, to the best knowledge of the author. References and acknowledgements to other researchers have been given as appropriate.

Chapter 1

Introduction

Contents

1.1 Overview

Underwater communication networks have recently received much attention due to the applications in oceanographic data collection, pollution monitoring, offshore exploration, disaster prevention, mine reconnaissance, tactical surveillance, etc [2] [3]. In underwater environments, high frequency radio waves, widely used in wireless communications in the air, are highly attenuated and absorbed by the conductive salty water. The propagation range of radio waves is highly restricted, depending on the frequencies. Only extra low-frequency waves ($30 - 300$ Hz) propagate long distance in underwater [2], which require large antennas. Optical waves do not suffer from high attenuation, however, the high scattering effect in the water and the difficulty in designing narrow laser beams restrict the application [2]. Therefore, transmitting signals using acoustic waves is the most popular

physical layer technology for underwater networks.

The Underwater acoustic channel is considered as one of the most difficult channels for communications [4]. Underwater sound speed (typically 1500 m/s) is five orders lower than the speed of light (3×10^8 m/s). Comparing to radio communication systems, the propagation delays between sources and receivers are relatively large in underwater communication systems. This increases the difficulty in time synchronization and designing network protocols [5]. The sound speed is a function of environmental parameters such as water density, temperature and salinity, which vary in time and space [6]. Therefore, different ocean areas have different channels at different time. Generally, underwater acoustic channels have relatively large delay spread, on an order from ten to hundred of symbol durations, depending on the environment and the distance between the source and the receiver [7] [8]. This requires a large number of taps to represent the channel impulse response and results in high inter symbol interference. Besides, underwater acoustic channels may be sparse [7] [8] [9] [10], e.g., among the relatively large number of channel taps, only a small number of taps have non-zero coefficients. Most of the tap coefficients are close to zero.

Underwater acoustic signals may suffer from significant Doppler effects. Movements of underwater transmitters/receivers are almost unavoidable and are time-varying due to the motion of the ocean. The relatively low speed of sound and possible wide-band modulation makes the Doppler shifts several orders higher than those of radio signals [11]. Different multipath arrivals have different Doppler shifts due to their different propagating directions, resulting in Doppler spread. This Doppler spread may be significant when the differences in the angles of directions are large.

Signal processing techniques developed for radio channels may not perform equivalently well in underwater acoustic systems, due to the features of underwater acoustics described above. In this thesis, we develop signal processing techniques that can be used in underwater acoustic networks.

The performance of underwater acoustic communication systems is heavily dependent on the propagation environment. For assessing the communications performance, sea experiments are required. Although sea experiments are ultimate means to assess the per-

formance, they are difficult to conduct and very expensive. In some situations, instead, simulation of the propagation channel can be used. The simulation also has other advantages compared to experiments. E.g., due to highly dynamic time and space variability of the underwater environment, it is difficult to guarantee similar experimental conditions when comparing different systems. It is also difficult to provide reliable monitoring of the environment, and thus give valuable interpretation of experimental results. Furthermore, ocean experiments with multiple users (such as in underwater communications networks) are even more complicated. On the other hand, computer simulation can provide exactly the same propagation conditions when investigating different systems, precise monitoring of the environment, and modeling communication networks with multiple users. However, this is only possible if the simulator is capable of providing results similar to that observed in sea experiments. Thus, an efficient approach to simulate underwater acoustic signal transmission is highly desirable [2, 12, 13]. In this thesis, a low-complexity simulator is developed for generating signals in the scenarios with moving transmitter/receiver.

For underwater networks, knowledge of positions of communication nodes is important. For instance, the propagation delay of signals among the nodes can be estimated according to the positions, which is helpful for synchronization. In a sensor network, the data collected by the nodes may be useless without the knowledge of node positions. Due to the invalidity of plane wave model for underwater acoustic waves, matched filed (MF) processing, a beamforming technique using full propagation model of acoustic waves [14], has been developed and can provide accurate localization results [6, 15–19]. Since there are possible multiple nodes in a network, in this thesis, we develop and investigate a multi-source localization technique based on MF processing. As the knowledge of environmental parameters, such as the sound speed profile (SSP), are required for MF localization techniques, we also investigate an MF inversion (MFI) problem for estimating the uncertain SSP.

Channel estimation is a very important task for communication receivers. Accurate estimation of underwater acoustic channels is a challenging task since the channels vary in time, have a large number of coefficients to be estimated, and can be sparse [10]. To address these problems we adopt the basis expansion model (BEM) [20–23] to track the time variations of the impulse response; we develop a low-complexity technique for dealing with channels with a large number of taps. The channel sparsity is taken into

account by determining the delay spread and number of non-zero taps according to the source/receiver positions.

1.2 Contributions

Major contributions of this thesis are summarized as follows.

- A low-complexity simulator for modeling underwater acoustic transmission with moving transmitter/receiver has been developed. The motion induced channel time variations are modeled by computing the time-varying impulse response according to the transmitter/receiver moving trajectories and using an acoustic field computation method. Different acoustic field computation methods can be used in the simulator. The complexity of the simulator is kept low by adopting local-spline interpolation and using a delay adjustment method for reducing the rate of field computation. The simulator can be implemented in a recursive manner and, thus, can be used for modeling signal transmission for arbitrary long trajectories. The developed simulator has been verified by comparing the simulated data with data from real ocean experiments. In particular, applying simulated data to an OFDM modem shows similar performance with that obtained from the data of a deep water experiment.

- An MF multi-source localization technique has been developed. The proposed technique locates sources by solving a set of basis pursuit de-noising (BPDN) problems corresponding to a set of source frequencies. A grid refinement method has been applied to reduce the complexity in solving the BPDN problems. The proposed technique outperforms other MF localization techniques in resolving sources positioned closely to each other, tolerance to noise and capability of locating multiple sources. The performance of the technique has been verified using simulated and real experimental data.

- An efficient technique combining the homotopy approach and coordinate descent (CD) search has been developed for solving complex-valued BPDN problems. This technique has been used to solve BPDN problems for source localizations and

identify non-zero taps when estimating time-varying sparse channels.

- Several global optimization algorithm based on simulated annealing (SA) are compared. An algorithm combining fast simulated annealing (FSA) method and downhill simplex (DHS) method has been applied for solving an MFI problem to estimate uncertain sound speed profile (SSP) in a deep water scenario. Accurate MF localization results have been demonstrated using the estimated SSP.

- A low-complexity BEM based channel estimator using local splines had been developed for estimating channels with large delay spread. In dealing with time-varying multipath Rayleigh fading channels, the proposed estimator has been shown to have better estimation accuracy than the maximum likelihood (ML) estimator and have close estimation performance to the minimum mean square error (MMSE) estimator.

- The proposed channel estimator has been applied to estimate time-varying sparse underwater acoustic channels. The sparsity of the channel has been taken into account for improving the estimation accuracy and reducing the complexity.

1.3 Thesis outlines

The rest of this report is separated into the following chapters.

- Chapter 2: Fundamental Techniques: Computation of Underwater Acoustic Field

  In this chapter, fundamental techniques used throughout this thesis are introduced. Firstly, the propagation model of underwater acoustic waves as well as three different acoustic field computation methods, the normal mode method, fast field program, and Gaussian beam tracing method are described. The programs corresponding to the three methods, KRAKEN, SCOOTER and BELLHOP, as well as their environmental parameter settings are also presented with examples.

- Chapter 3: Modeling of time-varying underwater acoustic channels

  This chapter presents a simulator developed for modeling transmission of underwater acoustic signals with moving transmitter/receiver. The motion induced channel

time variations are modeled by sampling the trajectory at the signal sampling rate and calculating for each position the channel impulse response from acoustic field computation. To reduce the complexity, the time-varying impulse response is generated by interpolating over a set of field-computed channel impulse responses of a relatively small number of sampling points on the trajectory. The computation of impulse response for a given bandwidth based on different acoustic field computation methods is described. An approach for adjusting time-varying delays of multipath arrivals is presented for reducing the sampling rate of the trajectory. A simple criteria for setting the sampling interval of the trajectory is also presented. We finally compare data generated by the simulator to that obtained from two ocean experiments.

- Chapter 4: Underwater localization of multiple sources using BPDN

  This chapter presents an MF multi-source localization technique. The proposed technique exploits formulation of the localization problem in terms of sparse representation of a small number of source positions among a much larger number of potential positions. The sparse representation is formulated as the BPDN problem for complex-valued variables. An efficient algorithm combining the homotopy approach and CD search is derived for solving the BPDN problem. The homotopy algorithm is modified for multi-frequency signals to provide more reliable localization results. Numerical results obtained from the proposed localization technique, using simulated data and real experimental data, are compared with the results obtained from other MF techniques.

- Chapter 5: Estimation of SSP using MFI

  In this chapter, we investigate the estimation of uncertain SSP by solving an MFI problem. Several global optimization methods are compared and an efficient adaptive global optimization method, combining the local optimization method DHS and the global optimization method FSA, is used to solve the inversion problem for estimating SSP in a deep water scenario.

- Chapter 6: Estimation of Time-varying Sparse Underwater Acoustic Channels

  This chapter investigates low complexity channel estimation for sparse underwater acoustic channels. An iterative channel estimator using cubic B-splines for approximating the time-varying impulse response is developed for reducing the high

complexity caused by large delay spread. The complexity of the proposed estimator is analyzed, comparing with that of the ML BEM channel estimator. A method for identifying non-zero channel taps by solving a BPDN problem using the homotopy algorithm developed in Chapter 4 is described. The iterative estimator is firstly applied for estimating short time-varying multipath Rayleigh fading channels to justify its performance. Then the iterative estimator is applied for estimating time-varying sparse underwater acoustic channels, generated by the simulator developed in Chapter 3, and combined with the non-zero tap identification method.

1.4 Notations

In this thesis, we will use the following notations unless otherwise specified. The symbol j is an imaginary unit $j = \sqrt{-1}$. $\Re\{\cdot\}$ denotes the real part of a complex number. Matrices and vectors are denoted by capital and small bold fonts, respectively, e.g. $\mathbf{A}$ is a matrix and $\mathbf{b}$ a vector. Elements of the matrix and vector are denoted as $[\mathbf{A}]_{m,n}$ and $[\mathbf{b}]_n$, respectively; $\mathbf{A}^{(n)}$ denotes the nth column of matrix $\mathbf{A}$. $(\cdot)^T$, $(\cdot)^H$, $(\cdot)^*$ denote the matrix transpose, complex conjugate transpose, and complex conjugate, respectively. $||\cdot||_p$ is the l_p norm. $\otimes$ denotes the Kronecker product. $E\{\cdot\}$ denotes the statistical expectation operator.

1.5 Publication List

Journal Papers

1. T. Chen, Y. V. Zakharov and C. Liu, "Low-Complexity Channel-Estimate Based Adaptive Linear Equalizer", IEEE Signal Processing Letters, vol. 18, no. 7, pp. 427–430, 2011.

2. C. Liu, Y. V. Zakharov and T. Chen, "Doubly-Selective Underwater Acoustic Channel Model for Moving Transmitter/Receiver", accepted by IEEE Transactions on Vehicular Technology.

3. C. Liu, Y. V. Zakharov and T. Chen, “Broadband underwater localization of multiple sources using basis pursuit de-noising”, accepted by IEEE Transactions on Signal Processing.

4. T. Chen, C. Liu, and Y. V. Zakharov, “Source localization using matched phase matched-field processing with phase descent search”, accepted by IEEE Journal of Oceanic Engineering.

Conference Papers

1. C. Liu, T. Chen and Y. V. Zakharov, “Source localization using sparsity based iterative adaptive beamforming ”, in Proc. of the 10th European Conference on Underwater Acoustics (ECUA10), Istanbul, Turkey, 2010, pp. 604–610.

2. C. Liu, T. Chen and Y. V. Zakharov, “Matched field inversion for sound speed profile in a deep water environment by using simplex simulated annealing”, in Proc. of the 10th European Conference on Underwater Acoustics (ECUA10), Istanbul, Turkey, 2010, pp. 598–602.

3. T. Chen, C. Liu and Y. V. Zakharov, “Matched-phase coherent broadband matched-field processor using phase descent search ”, in Proc. of the 10th European Conference on Underwater Acoustics (ECUA10), Istanbul, Turkey, 2010, pp. 590–595.

4. C. Liu, T. Chen and Y. V. Zakharov, “Broadband underwater source localization by solving basis pursuit de-noising using coordinate descent search”, in Proc. of IEEE 7th ISWCS, York, UK, 2010, pp. 1–5.

5. C. Liu, T. Chen and Y. V. Zakharov, “Modeling of time-varying underwater acoustic channels ”, presented in 4th Conference on Underwater Acoustic Measurements: Technologies and Results, Kos, Greece, 2011.

Chapter 2

Fundamental Techniques: Computation of Underwater Acoustic Fields

Contents

In this chapter, fundamental techniques used in this thesis for computing underwater acoustic field are briefly introduced.

2.1 Underwater acoustic field computation methods

Underwater acoustic waves suffer from refraction due to the change of sound speed in water. The plane wave model, which is usually used for modeling the propagation of radio waves, does not hold for underwater acoustic waves. In this section, the underwater acoustic wave equation and three methods for computing acoustic fields are briefly introduced.

For range independent environment (the environmental parameters such as the sound speed and the density are only functions of depth) and an isotropic source $s(t)$, the acoustic field pressure p at the measured depth z and horizontal distance r from the source can be calculated by solving the acoustic wave equation given by [24]:

$$\bigtriangledown\cdot\left(\frac{1}{\rho}\bigtriangledown p\right)-\frac{1}{\rho c^2(z)}\frac{\partial^2 p}{\partial t^2}=-\frac{s(t)}{r}\delta(z-z_s,r), \tag{2.1}$$

where ρ is the density of water, $c(z)$ the sound speed at depth z, z_s the source depth, and $\delta(z-z_s,r)$ is the Dirac delta function.

With known environmental parameters such as SSP, water density and the source/receiver positions, the acoustic field can be calculated by solving the wave equation in (2.1). There are various methods for solving the equation. The method of solving the wave equation is not the subject of interest in this thesis. However, the basic concept behind each method is worth to know since it is useful to understand how to use the program for computing the acoustic field and how to set the parameters. In the following, three popular methods, namely, the normal mode method, the Gaussian beam tracing method and the Fast field program (FFP) method, are briefly introduced.

2.1.1 Normal mode method

The underwater environment can be considered as a waveguide, where the sea surface and bottom are treated as boundaries. Therefore, the normal mode method, which is widely used for solving waveguide problems, can be used to calculate the acoustic field. For simplicity, a single-frequency source is considered, with a signal $s(t)$ represented as

$$s(t)=e^{-j\omega t}, \tag{2.2}$$

where the amplitude of the source signal is assumed to be 1, ω is the circular frequency and t is the time. The resulting acoustic field can be represented as a similar harmonic:

$$p=p(r,z)e^{-j\omega t}, \tag{2.3}$$

where $p(r,z)$ is the complex amplitude of the field pressure. To start with, we can rewrite the wave equation in (2.1) in cylindrical coordinates and factor out the harmonic [1]:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial p(r,z)}{\partial r}\right)+\rho(z)\frac{\partial}{\partial z}\left(\frac{1}{\rho(z)}\frac{\partial p(r,z)}{\partial z}\right)+\frac{\omega^2}{c^2(z)}p(r,z)=-\frac{\delta(z-z_s)\delta(r)}{2\pi r}. \tag{2.4}$$

Using the technique of separation of variables, letting $p(r,z) = Z(z)R(r)$, and assuming a perfectly rigid bottom condition, we obtain the following mode equations [1]:

$$\rho(z)\frac{d}{dz}\left(\frac{1}{\rho(z)}\frac{dZ(z)}{dz}\right) + \left(\frac{\omega^2}{c^2(z)} - k^2\right)Z(z) = 0, \tag{2.5}$$

$$Z(0) = 0, \tag{2.6}$$

$$\frac{dZ(D)}{z} = 0, \tag{2.7}$$

where D is the bottom depth. The solution of these equations are a set of Green functions (modes). The acoustic field can be expressed by:

$$p(r,z) = \sum_{m=1}^{\infty} R_m(r)Z_m(z), \tag{2.8}$$

where $Z_m(z)$ is the mth mode function and $R_m(r)$ are the corresponding coefficients. Equations in (2.5) and (2.8) indicate a basic procedure for computing the acoustic field. The mode functions $Z_m(z)$ are firstly computed according to the sound speed profile $c(z)$ and the density $\rho(z)$. The acoustic field is then calculated according to the mode functions $Z_m(z)$ and the positions of the source and receiver.

2.1.2 Fast Field Program (FFP) method

FFP treats the wave equation in a different way from that of the normal mode method. In FFP, the Fourier-Bessel transform is applied to the wave equation in (2.1), which is given by [24]:

$$\hat{p}(k,z,t) = \int_0^{\infty} p(r,z,t)J_0(kr)r\mathrm{d}r, \tag{2.9}$$

where $J_0(kr)$ is the zero-order Bessel function and k is a parameter related to the wavelength. We then obtain from (2.1) [24]:

$$\frac{\partial}{\partial z}\left(\frac{1}{\rho}\frac{\partial\hat{p}(k,z,t)}{\partial z}\right) - \frac{k^2}{\rho}\hat{p}(k,z,t) - \frac{1}{\rho c^2(z)}\frac{\partial^2\hat{p}(k,z,t)}{\partial t^2} = -s(t)\delta(z-z_s). \tag{2.10}$$

For a sequence of k values, (2.10) is solved. The acoustic field is then calculated by the inverse Fourier-Bessel transform:

$$p(r,z,t) = \int_0^{\infty} \hat{p}(k,z,t)J_0(kr)k\mathrm{d}k. \tag{2.11}$$

Similar to the normal mode method, FFP has two steps to compute the acoustic field. A series of functions $\hat{p}(k,z,t)$ are computed according to the environmental parameters

as well as the depths of the source and the measured point. At the next step, the horizontal distance from the source to the measured point should be defined to compute the acoustic field.

2.1.3 Gaussian beam tracing method

Beam tracing algorithm is originated from and based on ray tracing method [25]. Tracing the evolution of a beam starts with solving the ray equation to obtain the central ray of each beam. The beam is then constructed around the central ray. For computing the acoustic field, a series of rays with different casting angles are generated. The acoustic field around each ray is computed by solving the wave equation, where the wave equation is usually converted to its equivalent form, a parabolic equation [25] [26]. It is known that high frequencies wave field propagates mostly along the rays [26], e.g., the acoustic field decays as the distance from a ray increases. Solutions of the parabolic equation show that the field pressure in a beam falls off in a Gaussian fashion as a function of the distance from the central ray [26].

For computing the acoustic field of a point source, a series of beams are generated from the source and traced in the water. The beam pattern can be defined by the casting angle interval of the rays from the source and the number of beams. The acoustic field at the receiver is then calculated by summing the contributions of each individual beam that cross the vicinity of the receiver.

2.2 Field computation programs

In Section 2.1, we have briefly introduced three different propagation models developed for computing the underwater acoustic field. In this section, the corresponding programs, which can be found in the acoustic toolbox [1], will be introduced. All the three programs are developed in FORTRAN. However, they have been compiled into executable files so that they can be easily used in MATLAB.

2.2.1 Normal mode program: KRAKEN

KRAKEN is a normal mode method based program for computing underwater acoustic field. It includes two core programs: the KRAKEN program and the FIELD program. The KRAKEN program computes the mode functions according to the environmental parameters. The FIELD program calculates the acoustic field using the mode functions computed by the KRAKEN. As mentioned in Section 2.1, to calculate the mode functions, the parameters describing the underwater environment should be defined. These parameters include the SSP (the sound speed at different depths), density of the water, characteristics of the bottom, etc. Besides, the source frequency should also be provided at this stage. For the KRAKEN program, a specific environment file 'ENVFIL' describing these parameters are required as an input. The outputs of KRAKEN are the mode functions, included in a 'MODES' file.

According to the mode functions, by defining the positions of the source and receiver, the acoustic field is computed by another core program FIELD. Additional parameters describing the number of modes involved in computing the acoustic field as well as the positions are included in a separate file 'FLDFIL'. The computed acoustic field is saved in an output file 'SHDIF'. A block diagram of the KRAKEN is provided in Fig.2.1.

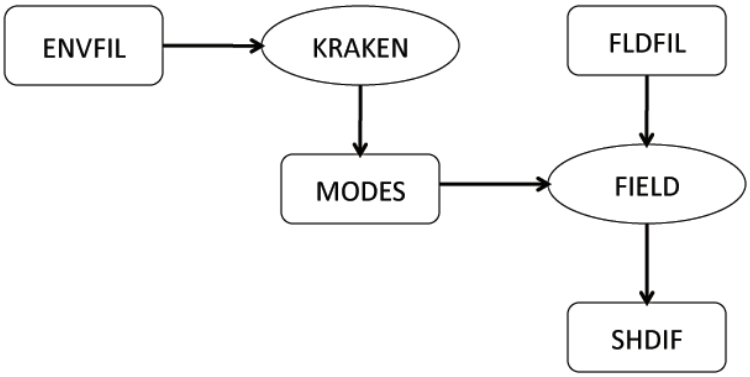


Figure 2.1: The structure of KRAKEN [1].

2.2.2 FFP: SCOOTER

SCOOTER is an FFP based program for computing the underwater acoustic field. Similarly to KRAKEN, there are two core programs, SCOOTER and FIELD, for computing

the functions $\hat{p}(k, z, t)$ and the acoustic field, respectively. Therefore, SCOOTER has a similar structure as that of KRAKEN. However, it should be noticed that additional information, including the depths of the source and receiver, should be provided in the 'ENVFIL'. In the acoustic toolbox, the positions of the source and receiver are defined in 'ENVFIL' so that the environmental file is compatible for SCOOTER and KRAKEN. The block diagram of SCOOTER is illustrated in Fig.2.2.

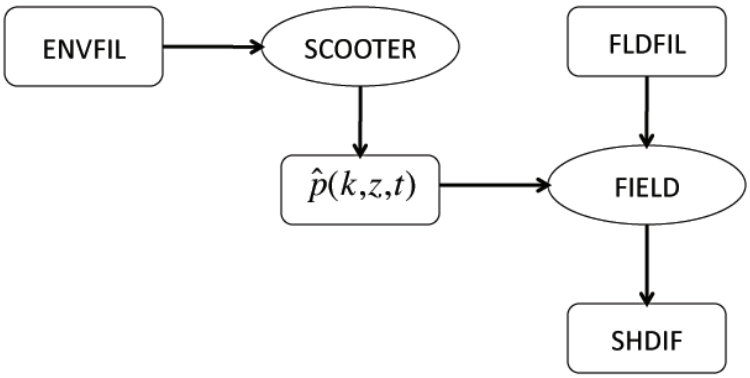


Figure 2.2: The structure of SCOOTER [1].

2.2.3 Beam tracing program: BELLHOP

BELLHOP is a Gaussian beam tracing based program for underwater acoustic field computation. It is convenient to show the evolution of the rays for a given environment. The evolution of the rays is included in a file 'RAYFIL'. Besides, the BELLHOP is also capable of providing the multipath arrivals at the receiver (included in 'A-DFIL'), with the propagation time of each arrival and the corresponding complex-valued amplitude, which are useful for computing the channel impulse response for a given bandwidth. Fig.2.3 gives a block diagram of BELLHOP.

When running the BELLHOP program, one of the three output files shown in Fig.2.3 is generated. There is a parameter in 'ENVFIL' describing the type of output. As mentioned in Section 2.1, the beam pattern should also be defined. This includes the interval of the emission angles, number of beams, the stepsize of tracing the rays, and the maximum range and depth of the tracing. All the parameters are included in additional lines in the 'ENVFIL' file, compared to that of KRAKEN and SCOOTER.

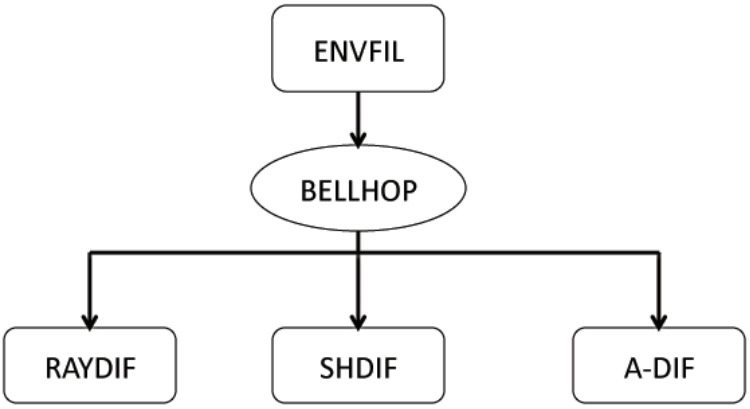


Figure 2.3: The structure of BELLHOP [1].

The three programs described above are all capable of providing accurate field computation when the environment parameters are properly set. However, for different scenarios, the accuracy and efficiency of the three programs are different. Generally, SCOOTER is suitable for computing the field of a short range from the source. The computation time increases when the distance between the source and receiver increases. KRAKEN is also able to calculate the field of short ranges, however, extra attention should be paid for setting the parameters to avoid the possible inaccuracy in computing the mode functions [1]. Therefore, as a rule of thumb, computing the field for distances within 10 water depth is carried out by SCOOTER. For larger distances, KRAKEN is used [1].

The beam tracing program BELLHOP has lower accuracy than KRAKEN. However, for KRAKEN, since the number of mode functions increases as the frequency increases, the computational complexity for high frequencies is high. Therefore, to save the computation time, BELLHOP is often chosen for dealing with high frequencies.

2.3 Simulation results

So far, we have introduced and described three different field computation programs. In this section, we will give examples of the parameter setting for the programs, which will be used in Chapter 5 for the inversion of SSP. Numerical results for field computation are also provided.

2.3.1 Parameter setting

A typical deep water SSP, the Munk profile [27], is used in this section. It has a sound speed $c(z)$ as a function of the depth z given by:

$$c(z) = 1500\left[1.0 + \epsilon\left(\frac{2(z-1300)}{1300} - 1 + e^{-\frac{2(z-1300)}{1300}}\right)\right], \tag{2.12}$$

where the quantity ϵ is taken to be $\epsilon = 0.00737$. The resulting SSP is plotted in Fig.2.4, where the bottom depth is 5000m.

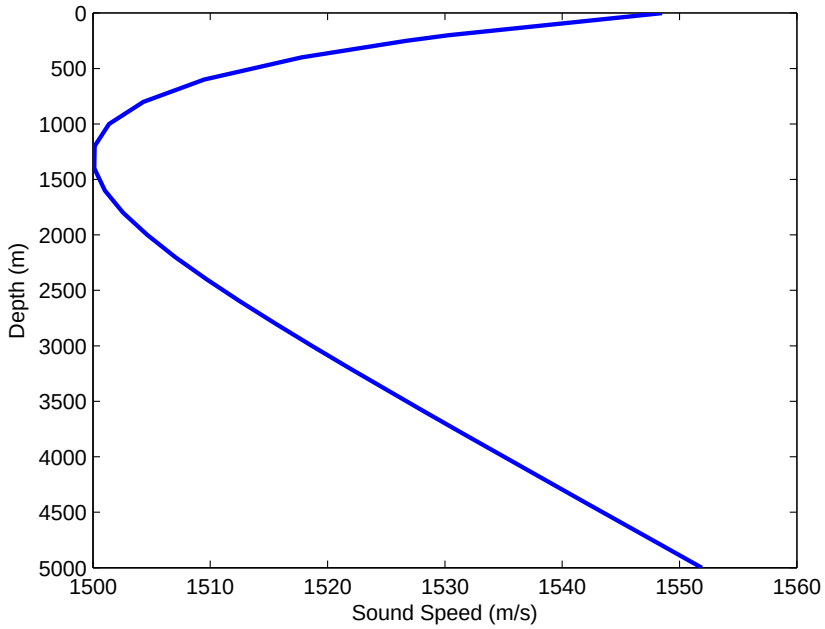


Figure 2.4: Munk Sound Speed Profile.

The 'ENVFIL' for the KRAKEN and the SCOOTER is listed in Table 2.1. The text after sybmol '!' on each line is the comment. As shown in the table, the first line is the title of the SSP. The second line is the source frequency in Hz, which is fixed when the computation is performed. The third line is the number of layers from the surface to the bottom boundary. For instance, in some underwater scenarios, the upper layer of the bottom is sand, while the deeper layer is rock. Then we can treat the rock layer as a perfect rigid boundary. For the upper layer, the SSP should be provided in the same way with that for the water column as an additional layer.

The forth line comes with some options about interpolation of the SSP, surface boundary condition and attenuation unit. In Table 2.1, the letters 'CVW' specify linear inter-

Table 2.1: Munk Environment File for KRAKEN and SCOOTER

Line		
1	'Munk Profile	! Title
2	50.0	! Frequency (Hz)
3	1	! Number of Layers
4	'CVW'	! Options: Interpolation, Surface boundary, Attenuation
5	0 0.0 5000	! NMESH, SIGMA(m), Z(NSSP)
6	0 1548.2 0.0 1.0	! z (m), $c(z)$ (m/s), $c_s(z)$ (m/s), $\rho(z)$ (mg/cm^3)
7	200.0 1530.29 /	
8	250.0 1526.69 /	
9	400.0 1517.78 /	
10	... ...	
11	5000.0 1551.91 /	
12	'A' 0.0	! Bottom options
13	5000.0 1600.0 0.0 1.0 /	
14	1400.0 1600.0	! Minimum and Maximum sound speed (m/s)
15	200.0	! Maximum range (km)
16	1	! Number of source depth
17	200.0	! Source depth (m)
18	10	! Number of receiver depth
19	100.0 1000.0	! Receiver depth (m)

Table 2.2: Field File for KRAKEN

Line		
1	/	! Title
2	'RA'	! Options for source type and mode theory
3	9999	! Number of modes to include
4	1	! Number of SSPs
5	0.0	! The range for each SSP (zero for range independent SSP)
6	101	! Number of receiver range
7	0.0 100.0 /	! Receiver range (km)
8	1	! Number of source depth
9	200.0 /	! Source depth (m)
10	10	! Number of receiver depth
11	100.0 1000.0 /	! Receiver depth (m)
12	10	! Number of receiver depth
13	0.0 /	! Receiver displacement: Zeros for perfect vertical array

polation for the SSP, free space above the surface, and the attenuation unit dB per wavelength. Line five defines the number of mesh points (NMESH) used for representing the SSP, the surface roughness and the bottom depth of the last point of the SSP Z(NSSP). Setting NMESH $= 0$ means the program itself will decide the number of mesh points. More details for the options can be found in [1].

Line six starts to define the SSP $c(z)$. The speed of shear wave $c_s(z)$ [1] can also be defined. Line twelve defines the bottom options including the bottom type and the bottom roughness. The letter 'A' means the bottom is treated as an acousto-elastic half space [1]. The depth, sound speed, density of the bottom are defined in line thirteen. The upper and lower limit of the sound speed in the program, and information of source/receiver positions are defined from the fourteenth line to the nineteenth line.

As described in Section 2.1, a field file is required for both KRAKEN and SCOOTER to compute the acoustic field. Table 2.2 and Table 2.3 give examples of the field file 'FIDFIL' for KRAKEN and SCOOTER, respectively. For the field file of KRAKEN, the source and receiver positions are provided. For the SCOOTER's field file, there is only the information of the receiver range from the source. More detail about the options can be found in [1].

Table 2.3: Field File for SCOOTER

Line		
1	'RP'	! Options for coordinates and the integration of $\hat{p}(k, z, t)$
2	0.0 100.0 101	! Maximum range (km), Minimum range (km), Number of ranges

Table 2.4: Additional Lines for BELLHOP

Line		
14	1	! Number of source depth
15	200.0 /	! Source depth (m)
16	10	!Number of receiver depth
17	100.0 / 1000.0	! Receiver depth (m)
18	101	! Number of receiver range
19	0.0 100.0 /	! Receiver range (km)
20	'A'	! Output option
21	1000	! Number of beams
22	-25.0 25.0 /	! Angle interval (degrees)
23	125.0 5500.0 102.0	! Stepsize in range (m), Maximum depth (m), Maximum range (km)

For the BELLHOP, the beam pattern is defined in the 'ENVFIL'. The settings of the source frequency, SSP, and etc are the same to those for KRAKEN and SCOOTER. Additional lines for BELLHOP, replacing from the fourteenth line in Table 2.1, are listed in Table 2.4. In the 20th line, the letter 'A' means the output is the delay arrival file 'ADFIL'. There are other options for controlling the output file. Further details can be found in [1].

2.3.2 Field computation results

With proper environment settings, the above three programs can provide similar field computation results. In this section, we use the Munk SSP to compare these three programs. A uniform sampling of $c(z)$ in depth with a step of 200 m is used to represent the SSP. An additional point at a depth of 250 m is used to provide better interpolation accuracy for the SSP. The source is fixed at the depth of 200 m. The range interval considered

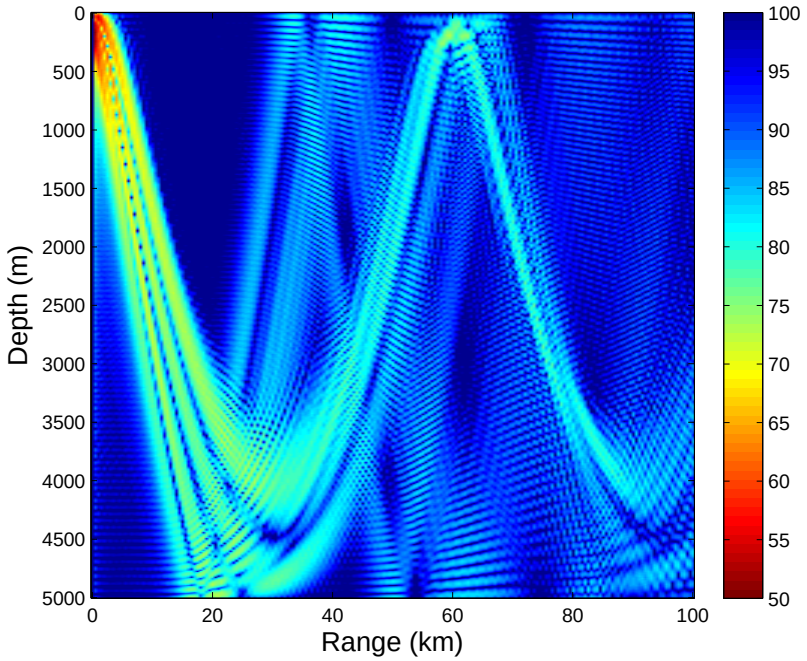
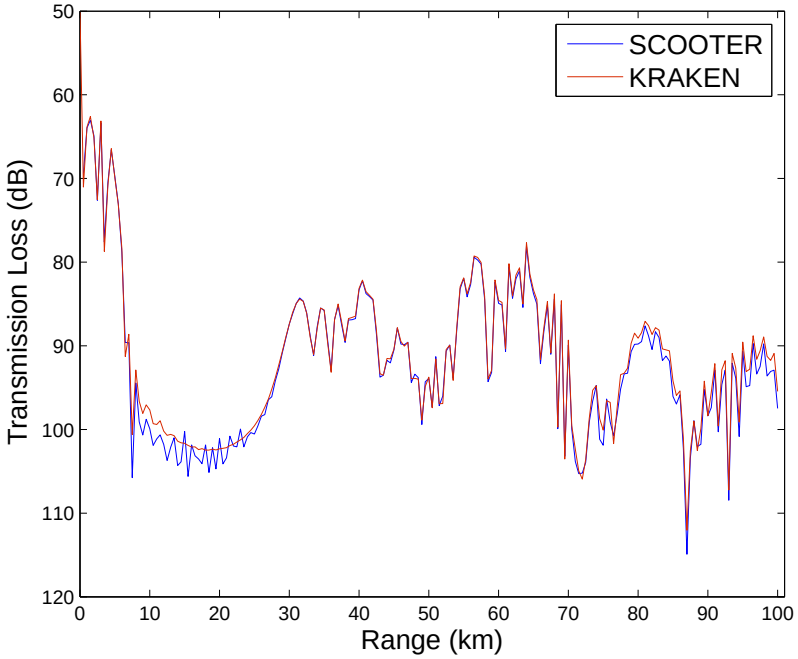


Figure 2.5: Field (transmission loss) computed by SCOOTER: the vertical bar shows the mapping of quantities (in dB) to corresponding colors.

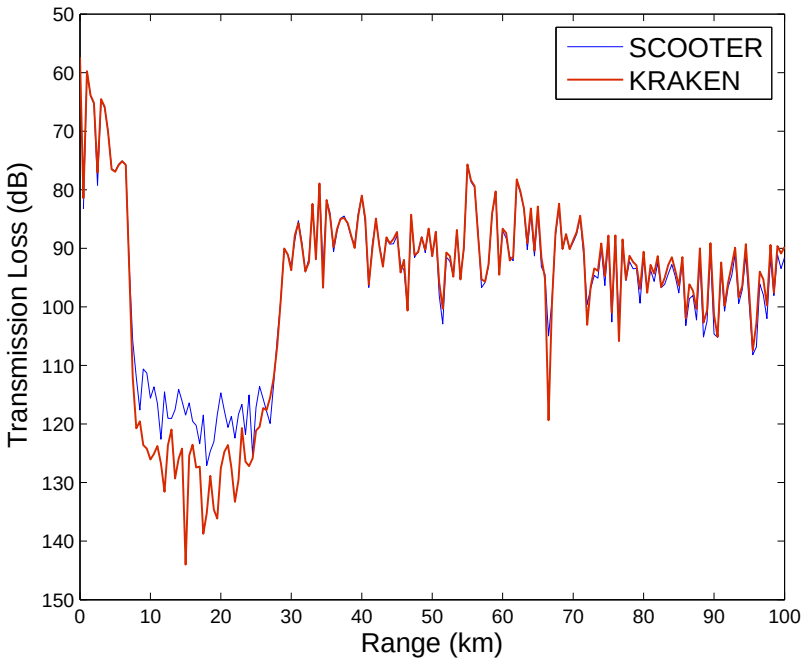
here is $[0, 100]$ km. Fig.2.5 shows the field computation results generated by SCOOTER at a frequency of 50 Hz. The acoustic field is represented as the transmission loss of the filed pressure at the measured point with respect to the field pressure at the source.

Fig.2.6 shows the field computation results generated by SCOOTER and KRAKEN for a receiver depth at 500 m. It is seen that for two different source frequencies (50 Hz and 1000 Hz), the acoustic field generated by the two programs are similar to each other. The most significant difference exists in range interval $[10, 30]$ km, where the receiver position falls into a shadow zone, as can be found in Fig.2.5.

For BELLHOP, to get accurate results, the beam tracing parameters should be set carefully. Fig.2.7 and Fig.2.8 give the results of several attempts for setting the beam patterns. The angle step of generating the beams is fixed at $0.1°$, e.g., for an angle interval $[-25°, 25°]$, the number of beams is 500. It is seen that, with increased tracing angles, the difference between the field computed by BELLHOP and that by SCOOTER is reduced. Compared with Fig.2.7(c) and Fig.2.7(d) for a frequency of 50 Hz (or Fig.2.8(c) and Fig.2.8(d) for a frequency of 1000 Hz), when the angle interval is large enough,



(a) 50 Hz



(b) 1000 Hz

Figure 2.6: Field (transmission loss) computed by SCOOTER and KRAKEN.

this difference does not change. This (systematic) difference between BELLHOP and SCOOTER is more significant than that between KRAKEN and SCOOTER.

From the results in Fig.2.7 and Fig.2.8, it is seen that to get accurate field computation from BELLHOP, the angle interval should be set large enough. This is due to the fact that the source is assumed to be isotropic, i.e., the signals are transmitted with the same power to different directions. In practice, the source always transmits signal in a smaller angle interval. An example of considering the limited angles for the source can be found in Chapter 3.

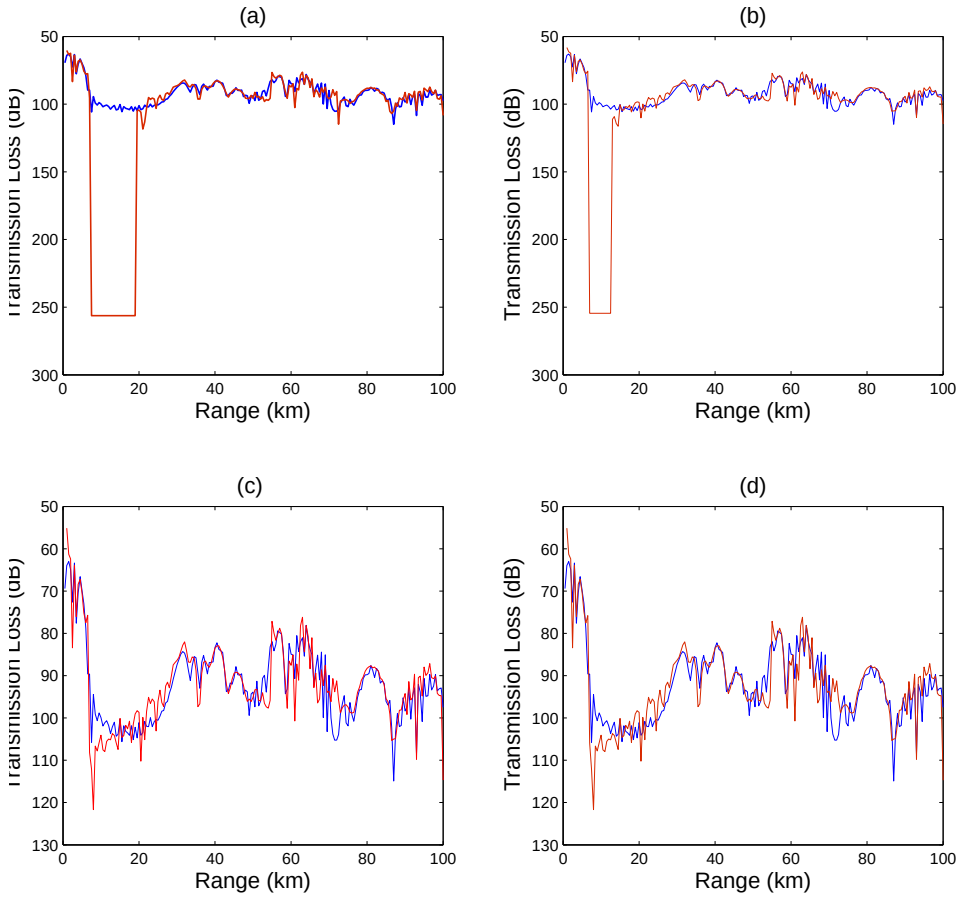


Figure 2.7: Field (transmission loss) computed by BELLHOP (red line) and SCOOTER (blue line) at a frequency of 50 Hz with angle intervals: (a) $[-25^\circ, 25^\circ]$; (b) $[-35^\circ, 35^\circ]$; (c) $[-55^\circ, 55^\circ]$; (d) $[-65^\circ, 65^\circ]$.

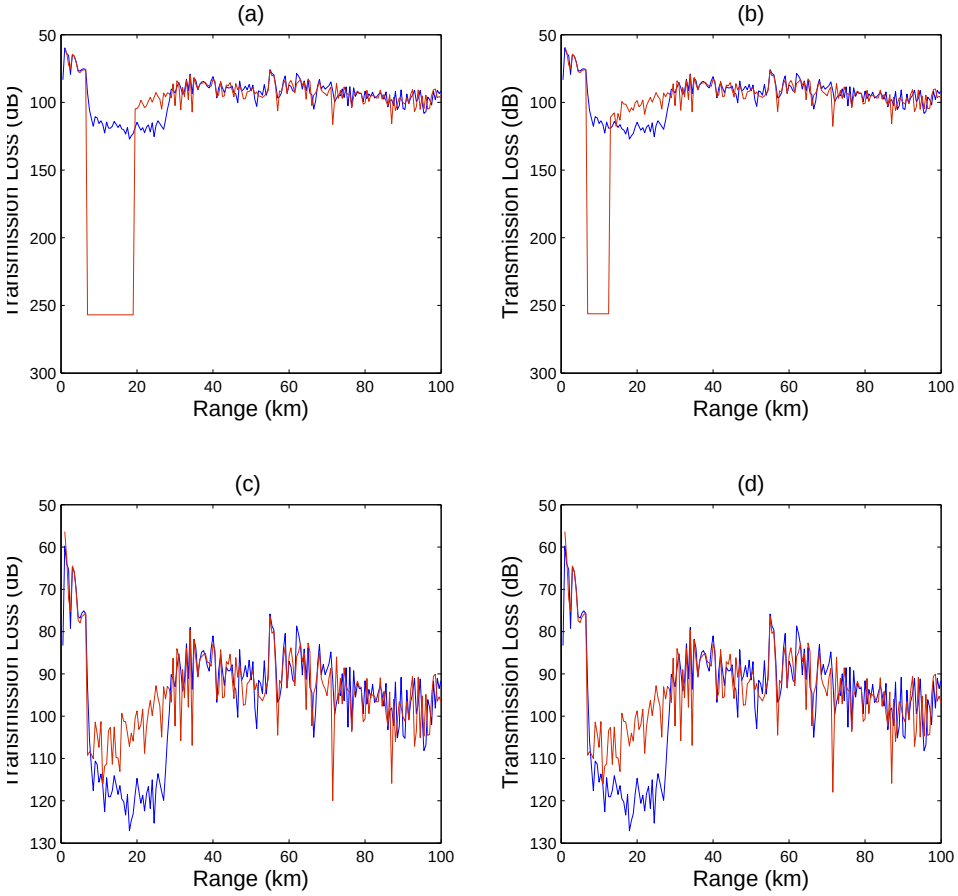


Figure 2.8: Field (Transmission loss) computed by BELLHOP (red line) and SCOOTER (blue line) at a frequency of 1000 Hz with angle intervals: (a) $[-25°, 25°]$; (b) $[-35°, 35°]$; (c) $[-55°, 55°]$; (d) $[-65°, 65°]$.

2.4 Summary

In this chapter, we have briefly introduced and described three different field computation methods for underwater acoustic waves, which are the normal mode method, the fast field program method, and the Gaussian beam tracing method. The corresponding programs for computing the acoustic field, KRAKEN, SCOOTER, and BELLHOP, are introduced and compared. Environmental parameter settings for each program have also been described using examples taken from the Munk sound speed profile. The three programs will be used for computing the acoustic field in the subsequent Chapters. Specifically, the normal mode program KRAKEN will be used for low frequency field computation

while the Gaussian beam tracing program BELLHOP will be used for high frequency computation. The fast field program SCOOTER will be used only for short ranges.

Chapter 3

Modeling of Time-varying Underwater Acoustic Channels

Contents

3.1 Introduction

Two important phenomena that affect the performance of underwater acoustic communications are the multipath propagation and the Doppler effect [2, 12, 28, 29]. The multipath propagation makes the channel to be frequency-selective and the Doppler effect makes the channel to be time-selective. Thus, the underwater acoustic channel is doubly-selective. Although, time-varying multipath channel models are widely used to analyze radio communication systems (see [30–32] and the references therein), they are not directly applicable to underwater acoustic communications [12, 33]. For radio communications, the Jakes' model combined with a set of standard power delay profiles with fixed multipath

delays is often considered to be useful for studying the system performance [34]. The important feature of the underwater acoustic channel is the fast variation of multipath delays due to a low speed of sound. As a result, the signal distortion caused by the Doppler effect in underwater channels includes the time compression/dilation that is different for different multipath components (see [11, 12, 29, 35–37] and references therein). Another important consideration is that a particular sea area can provide specific propagation conditions that should be taken into account when studying underwater acoustic systems. To account for the phenomena in the channel model, the time-varying channel impulse response needs to be computed for a specific sea area.

Time variations of the underwater acoustic channel are caused by many factors such as the transmitter/receiver motion [12], internal waves [38], surface waves [39, 40], and others. In this chapter, we focus on time variations caused by the transmitter/receiver motion. However, other phenomena can also be incorporated in the proposed model.

Extensive effort has been devoted in the literature to modeling underwater acoustic signal transmission. A channel simulator in [41] is based on a static channel impulse response computed by solving the underwater acoustic wave propagation equation using the normal mode method [1] or ray tracing method [25]. This, however, does not account for transmitter/receiver movement; thus, the Doppler effect is not considered. A simulator in [42] models fluctuations of the amplitude and phase of eigenpaths (with fixed delays) as random processes. However, it is unclear how statistics of the random processes should be defined depending on a particular sea area. Also, this model does not consider the delay variations, and therefore cannot accurately model the Doppler effect. The simulator proposed in [33] does incorporate the Doppler effect by introducing different frequency shifts in different eigenpaths; it then uses some statistical model for varying the multipath amplitudes. However, it does not model the time-varying multipath delays and it is unclear how to define statistics of the amplitude variations.

A general approach for simulation of signal transmission is to compute the channel impulse response for transmitter/receiver positions corresponding to all sampling instants of the signal. In [40], a method was developed for generating the channel impulse response by interpolating eigenpaths pre-computed at a regular space grid in the area surrounding the transmitter/receiver trajectory. However, for complicated and/or long-time movement,

where the number of grid points and the number of signal samples is large, the complexity of this approach is high.

In this chapter, we propose a different approach for modeling the underwater signal transmission that has a relatively low complexity and can be used together with different field computation methods. In this approach, the trajectory is sampled at a low rate (much lower than the signal sampling rate). At every trajectory sampling instant, a *waymark* impulse response is computed using a field computation method, e.g. the normal mode or ray tracing method. Using the waymark impulse responses and adjusting the multipath delays, the time-varying impulse response is recovered using local splines. The signals are then modeled by compensating for the adjusted delays and convolving the transmitted signal with the time-varying impulse response. The proposed approach employs recursive (in time) computations, thus allowing modeling the signal transmission for arbitrary long trajectories.

The rest of this chapter is organized as follows. In Section 3.2, the proposed approach is described. In Section 3.3, it is applied to model signals in a shallow water experiment. In Section 3.4, the proposed approach is used for modeling signals in a deep water experiment. Finally, Section 3.5 draws conclusions.

3.2 Underwater acoustic channel simulator

In a time-varying channel, the noise-free signal at a receiver is described as a convolution

$$y(t) = \int_{-\infty}^{\infty} h(t,\tau)s(t-\tau)d\tau, \;\; t \in [0, T_s], \tag{3.1}$$

of the channel impulse response $h(t,\tau)$ and the source signal $s(t)$, where T_s is the signal duration. This is the most general description of linear time-variant systems. In underwater acoustic communications, different variants of the description are used. E.g., for a channel with discrete multipath components, one can use the model [34, 35, 43]

$$h(t,\tau) = \sum_{p=1}^{L} A_p(t)\delta(\tau - \tau_p(t)) \tag{3.2}$$

where $A_p(t)$ and $\tau_p(t)$ are time-varying amplitudes and delays of L multipath components, and $\delta(t)$ is the Dirac delta function. If $\tau_p(t) = \tau_p - at$, we arrive at a channel model

that introduces compression/dilation with a compression factor a to the received signal; this model was adopted in [35] to describe non-uniform Doppler shifts of subcarriers in an OFDM system. If $\tau_p(t) = \tau_p - a_p t$, we obtain a model with different compression factors a_p for different multipath components as used in [43] to describe channel distortions introduced in OFDM signals. More complicated dependencies of delays $\tau_p(t)$ on time t allow representing time-varying Doppler shifts of signals. To describe macro- and micro-paths as well as the individual Doppler compression for different macro-paths, the following underwater acoustic channel model is used [44–46]

$$y(t) = \sum_{p=1}^{L} \int_{-\infty}^{\infty} s(\eta_p t - \tau) h_p(\tau - \tau_p) d\tau \tag{3.3}$$

where $\eta_p = 1 + a_p$ and $h_p(\tau)$ are impulse responses describing different micro-path structures for different discrete multipaths (macro-paths). This model is equivalent to (3.1) with

$$h(t, \tau) = \sum_{p=1}^{L} h_p(\tau - \tau_p(t)) \tag{3.4}$$

and $\tau_p(t) = \tau_p - a_p t$. It is worth to notice that the model (3.4) can easily incorporate the frequency dependent absorption, as opposed to the model (3.2).

In discrete-time form, the received signal is given by

$$y(nT) = \sum_{i=0}^{I-1} h(nT, iT) s(nT - iT), \;\; n = 0, \ldots, N-1, \tag{3.5}$$

where T is the signal sampling interval, I the number of channel taps, and $N = T_s/T$. For modeling the time-varying channel, the impulse response $h(t, \tau)$ should be known for every sampling instant $t = nT$.

Generating the impulse response for a particular point at the trajectory is based on the acoustic field computation, which is typically of high complexity. Therefore, generating $h(nT, \tau)$ for all n would make the simulation of long-time transmission (i.e. for high N) complicated. In this section, a method of modeling underwater acoustic signal transmission is proposed that allows reduction in the complexity. For the sake of clarity, we will concentrate on scenarios where the receiver is static and the transmitter is moving.

3.2.1 Generating the channel frequency response from acoustic field computation

Different source/receiver positions will result in different propagation channels. The motion induced channel time variations can be modeled by generating the channel impulse response for the trajectory points corresponding to signal samples. This can be done by defining the trajectory by the pair $[r(t), z(t)]$, where $r(t)$ is the horizontal distance between the transmitter and receiver and $z(t)$ is the transmitter depth at instant t, mapping the sampling instants $t = nT$ to positions $[r(nT), z(nT)]$, and finding the channel frequency response from acoustic field computation for a set of frequencies of interest. The channel impulse response can then be obtained from the channel frequency response by the inverse Fourier transform.

At a frequency ω_k, the channel frequency response $p(nT, \omega_k)$ can be computed by solving the wave equation that governs the propagation of underwater acoustic waves. The acoustic wave equation is defined by a set of environmental parameters, such as the water density ρ, the sound speed $c(z)$ at depth z, the source depth z_s, the receiver depth z, the horizontal distance to the source r, and some others [1]. With the knowledge of the environmental parameters, different methods can be adopted for computing the acoustic field; the most popular are the normal mode method [1] and the ray tracing method [25].

In the normal mode method, assuming that the environment is range independent, the frequency response at a frequency ω_k is calculated as [1]

$$p(nT, \omega_k) = \sum_{q=1}^{Q(\omega_k)} R_q(\omega_k, r(nT)) Z_q(\omega_k, z(nT)), \tag{3.6}$$

where $Z_q(\omega_k, z)$ are the normal modes and $R_q(\omega_k, r)$ are expansion coefficients. The main drawback of this method is that the normal modes should be repeated for each frequency required; this is computationally expensive. Besides, the number of normal modes $Q(\omega_k)$ to be taken into account increases at higher frequencies [1].

The ray tracing method [25] is based on computing L eigenpaths starting from the source and hitting the receiver. The propagation delays $\tau_{nl}(\omega_k)$ and attenuations $a_{nl}(\omega_k)$ computed for the lth eigenpath at the position $[r(nT), z(nT)]$ are then used to compute

the channel frequency response:

$$p(nT, \omega_k) = \sum_{l=1}^{L} a_{nl}(\omega_k) e^{-j\omega_k \tau_{nl}(\omega_k)}. \tag{3.7}$$

Compared with the normal mode method, the ray tracing method is more efficient for dealing with high frequencies [1].

As the channel frequency response should be calculated for N points of the trajectory, assuming K frequencies of interest ω_k, the field computation (e.g. the normal mode or ray tracing method) should be repeated KN times. E.g., for modeling the deep water signal transmission for the experiment in Section 3.4 with $N \approx 10^7$ and $K \approx 10^3$, about 10^{10} field computations would then be required. This will result in high complexity. The complexity can be reduced assuming that the multipath arrivals can be approximately treated as frequency independent [**?**]. However, this still requires $N \approx 10^7$ field computations. In the following steps, we are aiming to reduce this complexity.

3.2.2 Generating the channel impulse response

The (real-valued) channel impulse response for the position $[r(nT), z(nT)]$ can be computed as

$$h(nT, iT) = \Re \left\{ \frac{2}{K} \sum_{k=0}^{K-1} p(nT, \omega_k) e^{j\omega_k iT} \right\}, \tag{3.8}$$

where $\omega_k = (\omega^- + k\delta\omega) \in [\omega^-, \omega^+]$, $\omega^- > 0$ and $\omega^+ > 0$, $\omega^+ = \omega^- + (K-1)\delta\omega$, $j = \sqrt{-1}$, $K = (\omega^+ - \omega^-)/(\delta\omega) + 1$, and $i = 0, \ldots, I-1$. In general, the frequency sampling interval $\delta\omega$ is determined from the maximum propagation delay $\tau_{\max}$:

$$\delta\omega \leq \frac{\pi}{\tau_{\max}}. \tag{3.9}$$

In underwater acoustic channels, if the distance between the transmitter and receiver is high, due to the low speed of sound, $\tau_{\max}$ will be high, and with the small $\delta\omega$ the number K of frequencies at which the acoustic field should be computed will also be high; this will result in high complexity of computing the frequency response.

As the impulse response $h(nT, \tau)$ is only non-zero for a relatively short delay interval

$\tau \in [\tau_{\min}, \tau_{\max}]$, and $\tau_{\min}$ is usually close to $\tau_{\max}$, i.e. $(\tau_{\max} - \tau_{\min})/\tau_{\max} \ll 1$, we can write

$$\tilde{h}(nT, iT) = \Re \left\{ \frac{2}{K} \sum_{k=0}^{K-1} p(nT, \omega_k) e^{j\omega_k (iT + \tau_{\min})} \right\}, \tag{3.10}$$

where $\tilde{h}(nT, iT) = h(nT, iT + \tau_{\min})$. Thus we can replace the filtering of a signal with the impulse response $h(nT, iT)$ by a filtering with the impulse response $\tilde{h}(nT, iT)$ and an extra signal delay $\tau_{\min}$. Thus, the number of the channel taps I can be reduced from $I \sim \tau_{\max}/T$ to $I \sim (\tau_{\max} - \tau_{\min})/T$. Moreover, the frequency step $\delta\omega$ is then calculated as

$$\delta\omega \leq \frac{\pi}{\tau_{\max} - \tau_{\min}}. \tag{3.11}$$

Compared with (3.9), (3.11) allows higher $\delta\omega$ and smaller K, thus resulting in lower complexity of computing the frequency response, especially when the distance is long so that $\tau_{\min}$ is close to $\tau_{\max}$, i.e. $(\tau_{\max} - \tau_{\min})/\tau_{\max} \ll 1$.

3.2.3 Waymarks and local spline interpolation

For low-frequency signals, the variation of the channel impulse response from one signal sample in time to another can often be considered slow. Therefore, the computation of the impulse response for every signal sample will be redundant, and the trajectory sampling interval T_w can be made much higher than the signal sampling interval T, i.e. $T_w \gg T$. Then, an interpolation procedure can be used for recovering the time-varying impulse response for all signal sampling instants. Specifically, we obtain $(M + 1)$ waymark impulse responses $h_m(iT) = h(mT_w, iT)$, where $i = 0, \ldots, I - 1$, $0 \leq mT_w \leq T_s$, $m = 0, \ldots, M$, and $T_w = T_s/M$.

In order to approximate time-variant channels, basis expansion models (BEMs) are widely used. The most often used BEMs are the Karhunen-Loeve (KL) functions [21,47], discrete prolate spheroidal (DPS) functions [20, 22], complex exponentials (CE) [22, 48, 49], and B-splines [23,50]. To describe channel time variations $h(t, iT)$ at a specific delay (tap) iT using a BEM we have the following approximation of $h(t, iT)$:

$$\hat{h}(t, iT) = \sum_{m=0}^{N_b - 1} c_{i,m} b_m(t), \tag{3.12}$$

where $b_m(t)$ are the N_b basis functions and $c_{i,m}$ are expansion coefficients for the ith channel tap. The use of the KL functions requires statistics of the path amplitude fluctuations to be known; however, we use a deterministic channel model. The KL, DPS, CE, as well as many other basis functions should be defined over the entire interval $[0, T_s]$. However, this would require: (a) computing and keeping in memory all waymark impulse responses over $[0, T_s]$ before expansion coefficients can be computed; (b) computing all expansion coefficients at once; and (c) keeping in memory and using all the expansion coefficients for computing the channel impulse response at any particular signal sampling instant. With high T_s, this results in high complexity and high memory consumption.

We use local splines that allow significant reduction in the complexity and memory requirement compared to the above basis functions. With the same number of basis functions we can achieve similar performance as that with an orthogonal basis [51, 52]. When using cubic B-splines, only four expansion coefficients and four basis functions are needed to compute the impulse response at any sampling instant. Furthermore, for computing an expansion coefficient, only a finite number (three in our case) of waymark impulse responses are needed. Thus, for computation of the signal at the output of the channel at any instant, we only need a few waymark impulse responses. E.g., for the cubic spline approximation that we are using, for any time instant only six waymark impulse responses are required. Besides, all basis functions are time shifted versions of a single B-spline that have a finite support ($4T_w$ for the cubic B-spline) which is usually much smaller than T_s. Firstly, this significantly reduces the modeling complexity. Secondly, this allows a recursive computation of the output signal with a reduced and fixed memory consumption independent of the signal duration.

Cubic splines provide good trade-off between the complexity and accuracy of approximation compared to other spline orders [53]. We use local cubic splines for our model, however other local splines can also be used [50, 54, 55]. The expansion (3.12) is then converted into [53]:

$$\hat{h}(t, iT) = \sum_{m=-1}^{M+1} c_{i,m} b(t - mT_w), \tag{3.13}$$

where we use $N_b = M + 3$ and $b(t)$ is the cubic B-spline given by

$$b(t) = \begin{cases} \frac{1}{6}\left(2 - \frac{|t|}{T_w}\right)^3 - \frac{2}{3}\left(1 - \frac{|t|}{T_w}\right)^3, & 0 \le |t| < T_w \\ \frac{1}{6}\left(2 - \frac{|t|}{T_w}\right)^3, & T_w \le |t| < 2T_w \\ 0, & \text{otherwise} \end{cases} \tag{3.14}$$

The expansion coefficients $c_{i,m}$ for the local splines are found as [50, 54]:

$$c_{i,m} = a_{-1}h_{m-1}(iT) + a_0 h_m(iT) + a_1 h_{m+1}(iT), \tag{3.15}$$

where $a_1 = a_{-1} = -1/6$ and $a_0 = 4/3$.

The spline interpolation can be represented in the matrix form. Let $\mathbf{h}_m = [h_m(0)\ h_m(T)\ \ldots\ h_m(IT - T)]^T$ and $\mathbf{c}_m = a_{-1}\mathbf{h}_{m-1} + a_0\mathbf{h}_m + a_1\mathbf{h}_{m+1}$, then

$$\hat{\mathbf{h}}(t) = \sum_{m=-1}^{M+1} \mathbf{c}_m b(t - mT_w).$$

Since the support of the cubic B-spline (3.14) is $4T_w$, for each time instant t only four spline coefficients $\mathbf{c}_m$ are required. Therefore, for $t = nT \in [mT_w, mT_w + T_w]$, we have

$$\hat{\mathbf{h}}(nT) = \mathbf{C}\mathbf{b}(nT), \tag{3.16}$$

where $\mathbf{C} = [\mathbf{c}_{m-1}\ \mathbf{c}_m\ \mathbf{c}_{m+1}\ \mathbf{c}_{m+2}]$, $\mathbf{b}(nT) = [b(\xi + T_w)\ b(\xi)\ b(\xi - T_w)\ b(\xi - 2T_w)]^T$, and $\xi = nT - mT_w$.

Note that if T_w is chosen to be a multiple of T, then the vectors $\mathbf{b}(nT)$ can be pre-computed for all possible ξ within one interval T_w and stored. Thus, recomputing this vector for each signal sample is not required. It is also seen that for each time instant nT, the spline interpolation has a recursive form with only 6 waymark impulse responses in the vicinity of the trajectory point $[r(nT), z(nT)]$ involved and no other information is required. This is beneficial when modeling long-time signal transmission.

3.2.4 Delay adjustment

Due to the low speed of sound, a small deviation of the transmitter/receiver position may result in significant deviation of the multipath propagation delays. With a high T_w, this

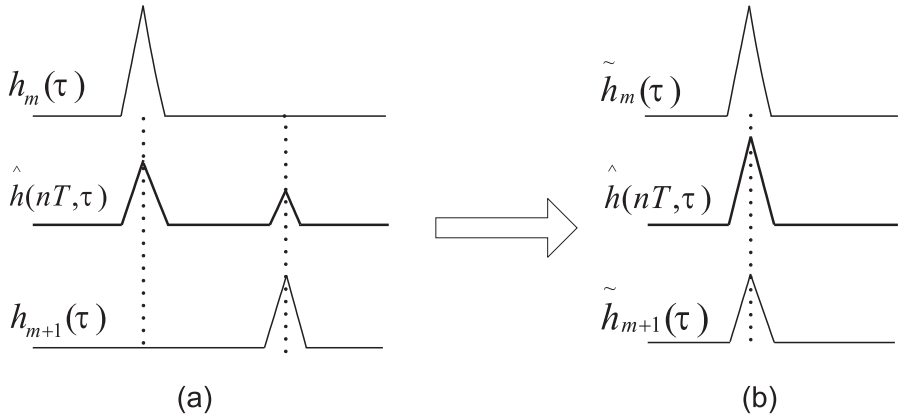


Figure 3.1: Example of interpolation of the channel impulse response: (a) without delay adjustment; (b) with delay adjustment.

can result in significant interpolation errors [13]. Fig.3.1(a) shows two adjacent waymark impulse responses obtained for a single path environment. We assume that the source is moving at a constant horizontal speed of 6 m/s and the channel delay varies in time as a linear function. Using $T_w = 125$ ms, i.e. the horizontal distance between two waymarks on the trajectory is 0.75 m, leads to increase in the delay between two waymarks by about 0.5 ms. When modeling signal transmission at a sampling rate of 10 kHz, this delay difference is as large as 5 signal samples. If we apply (for simplicity) linear interpolation for obtaining an impulse response at a time instant between the two waymarks, the result is a two-path impulse response $\hat{h}(nT, \tau)$ (instead of a true single path response) as shown in Fig.3.1(a). Thus, the multipath structure of the channel is destroyed.

To overcome this problem, we will try to compensate for the delay shifts between consecutive waymark impulse responses. Note that different multipath delays have different variations; therefore, in general, the precise compensation is infeasible. We can only achieve an approximate delay adjustment. This can be done in different ways. As an example, all the waymark impulse responses can be aligned with respect to the first multipath arrival. This however can encounter problems for some scenarios due to a possibility that the first arrival can disappear at some point of the trajectory. Another problem is that the first arrival can have the Doppler effect (i.e. the delay variation rate) quite different from that of other multipaths. We have found the following approach to aligning the time-varying impulse response efficient in many scenarios.

A delay shift Δ between two consecutive waymarks $m-1$ and m can be found by

Table 3.1: Delay adjustment

Input:
q^-, q^+, δ_{init}, Q, $p_{m-1}(\omega_k)$, $p_m(\omega_k)$, $k = 0, \ldots, K-1$
Initialization:
$\delta = \delta_{\text{init}}$
$q_{\max} = \arg\max_{q=q_{\text{lower}},\ldots,q_{\text{upper}}}\{J(q\delta)\}$
where $J(\Delta) = \left\lvert\sum_{k=0}^{K-1} p_m(\omega_k)p^*_{m-1}(\omega_k)e^{j\omega_k\Delta}\right\rvert^2$
$\Delta = q_{\max}\delta$
$J_1 = J(\Delta-\delta)$, $J_2 = J(\Delta)$, $J_3 = J(\Delta+\delta)$
Repeat Q times:
$\delta \leftarrow \delta/2$
if $J_3 < J_1$ then $J_3 = J_2$ and $\Delta \leftarrow \Delta - \delta$
else $J_1 = J_2$ and $\Delta \leftarrow \Delta + \delta$
$J_2 = J(\Delta)$

solving the following problem:

$$\Delta = \arg\max_{\Theta} J(\Theta) \tag{3.17}$$

where $J(\Theta) = \left|\sum_{k=0}^{K-1} p_m(\omega_k)p^*_{m-1}(\omega_k)e^{j\omega_k\Theta}\right|^2$. This problem is equivalent to the frequency estimation problem and it can be solved by many techniques. We have found that the dichotomous search similar to the technique in [56] is useful for this purpose. The corresponding algorithm is presented in Table 3.1.

The algorithm searches for the best match Δ for the two waymark frequency responses $p_m(\omega_k)$ and $p_{m-1}(\omega_k)$ in an interval $[q^-\delta_{\text{init}}, q^+\delta_{\text{init}}]$, where δ_{init} is an initial step size and q^+ and q^- are integers. The initialization $\delta_{\text{init}} = T/2$ is found to be a good choice. Assuming that the minimum (maximum) speed of the transmitter is $v_{\min}$ ($v_{\max}$), the values q^- and q^+ can be found from the inequalities:

$$q^+\delta_{\text{init}} > \frac{v_{\max}T_w}{c_0}, \quad q^-\delta_{\text{init}} < \frac{v_{\min}T_w}{c_0},$$

where $c_0 = 1500$ m/s.

The algorithm runs Q iterations in which the step-size δ is halved $\delta \leftarrow \delta/2$ and the solution Δ is rectified so that the final resolution is $\delta_{\text{init}}/2^Q$. A *waymark composite delay* is then found as

$$\tau_m = \tau_{m-1} + \Delta, \quad \tau_0 = \tau_{\min}.$$

The delay-adjusted waymark impulse response taking the composite delay into account is then generated as

$$\tilde{h}_m(iT) = \Re\left\{\frac{2}{K}\sum_{k=0}^{K-1} p_m(\omega_k)e^{j\omega_k(iT+\tau_m)}\right\}$$

i.e. $\tilde{h}_m(iT) = h_m(iT + \tau_m)$.

The delay-adjusted impulse response $\tilde{\mathbf{h}}_m = [\tilde{h}_m(0)\ldots\tilde{h}_m(IT-T)]^T$ will now replace $\mathbf{h}_m$ in the local interpolation described in Subsection 3.2.3. Fig.3.1(b) illustrates the benefit of the delay adjustment for the interpolation of the channel impulse response.

3.2.5 Waymark sampling period

We now discuss how the trajectory space sampling interval Δr and consequently the waymark sampling period T_w can be chosen. Firstly, one can use the trial and error approach. Several values of T_w can be tried and an interval providing a small enough error, e.g. the mean square error (MSE) between the interpolated and original (from the acoustic field computation) impulse responses as explained in Sections 3.3 and 3.4, can be used for the simulation.

Alternatively, the choice of Δr and T_w can be done based on the waveguide invariant theory [57, 58]. Specifically, we can write

$$\left|\frac{\Delta\omega}{\Delta r}\right| = \beta\left|\frac{\omega_{\max}}{r}\right|,$$

where $\Delta\omega$ is a frequency shift of a maximum $\omega_{\max}$ in the channel frequency response magnitude due to horizontal movement of a transmitter from distance r to $r + \Delta r$, and $\beta \approx 1$ is the waveguide invariant [57]. Due to the horizontal movement, the multipath delays are varying. We want to choose the range increment Δr so that a delay variation $\Delta\tau$ satisfies the condition: $\Delta\tau \ll T$. With such a choice fluctuations of the impulse response at fixed delays due to delay variations will be negligible.

Consider the following simplified scenario. Let at a distance r the channel have two equally strong multipath components of a unit amplitude separated by a delay τ. The magnitude of the channel frequency response is then given by $H(\omega) = 2 + 2\cos(\omega\tau)$.

The maximum of $H(\omega)$ is observed at frequencies $\omega_{\max} = 2\pi k/\tau$, where $k \geq 1$ is an integer. Due to the horizontal movement from r to $r + \Delta r$, the multipath delays change so that the magnitude of the frequency response is now $H_\Delta(\omega) = 2 + 2\cos(\omega(\tau + \Delta\tau))$. The maximum of $H(\omega)$ is now shifted to the frequency $\omega_{\max} + \Delta\omega = \frac{2\pi k}{\tau+\Delta\tau}$. From this equation, we find that $\frac{\Delta\omega}{\omega_{\max}} = -\frac{\Delta\tau}{\tau+\Delta\tau}$, and obtain

$$\Delta r = \frac{1}{\beta}\left|\frac{\Delta\omega}{\omega_{\max}} r\right| \approx \frac{\Delta\tau}{\tau + \Delta\tau} r.$$

We are interested in obtaining delay variations $\Delta\tau$ much smaller than the signal sampling interval T, i.e. Δr should be small enough to guarantee that $\Delta\tau \ll T$; thus

$$\Delta r \ll \frac{T}{\tau + \Delta\tau} r \approx \frac{T}{\tau} r.$$

For a fixed horizontal speed v_0 of the transmitter, we then arrive at the waymark sampling interval $T_w = \Delta r / v_0$. Although this is a simple scenario, we found that the relationship derived above is useful for choosing the waymark sampling interval for experimental scenarios. To apply this relationship in other scenarios, we note that τ is the multipath delay spread of the channel. Note that this choice is based on the assumption that the impulse responses at waymark points are aligned as explained in Subsection 3.2.4. In Sections 3.3 and 3.4, we show how this choice relates to the MSE.

3.2.6 Generating the channel output signal

For generating the channel output signal, we need for every signal sample instant nT:
a) compute the channel impulse response (this is done by local spline interpolation of the waymark impulse responses $\tilde{h}_m(iT)$ as described above);
b) compute and compensate for a composite delay $\hat{\tau}(nT)$;
c) convolve the input signal with the impulse response $\hat{\mathbf{h}}(nT)$.

For computing the composite delay we can again use the local spline interpolation similarly to (3.16):

$$\hat{\tau}(nT) = \boldsymbol{\theta}^T \mathbf{b}(nT),$$

where $\boldsymbol{\theta} = [\theta_{m-1}\ \theta_m\ \theta_{m+1}\ \theta_{m+2}]^T$ and $\theta_m = a_{-1}\tau_{m-1} + a_0\tau_m + a_1\tau_{m+1}$.

Compensation for the composite delay can be done as follows. A signal vector $\tilde{\mathbf{s}}(nT)$ of length I is computed with elements obtained by interpolation of the signal $s(t)$ at

instants $t = nT - iT - \hat{\tau}(nT)$, $i = 0, \ldots, I-1$. The interpolation can be done by using the local splines or any other method. The convolution then takes the form

$$y(nT) = \tilde{\mathbf{s}}^T(nT)\hat{\mathbf{h}}(nT).$$

The channel model is now summarized in Table 3.2.

The proposed simulator significantly reduces time for computing the acoustic field compared to the direct computation at the signal sampling rate f_s. For instance, computing the impulse response for one space point of the deep-water experiment, described in Section 3.4, using the ray tracing program BELLHOP [59] on a PC (4 GB memory, Intel CPU E8500, clocked at 3 GHz) takes approximately 3 s. Thus, generating a 100-s signal at the sampling rate $f_s = 12288$ Hz would take about $3.6 \cdot 10^6$ s or 42 days. In the proposed simulator using $T_w = 0.2$ s, for a 100-s signal, approximately 500 waymarks are required. Thus, computing the waymark impulse responses takes about 25 min. Note that the other computations for generating the signal, implemented in MATLAB, take also about 25 min. Thus, the whole computation of the 100-s signal takes about 50 min. Fig.3.2 shows the structure of the proposed simulator. The part of the simulator dealing with computations at the waymark rate $1/T_w$ (acoustic field, waymark impulse response, and composite delay computations) is separated from the part performing the signal processing at the signal sampling rate f_s. The latter can be implemented in real-time provided the waymark impulse responses and composite delays have been precomputed. The FIR filtering requires I MAC (multiply-accumulate) operations per sample. The delay compensation using the cubic local splines requires 3 MAC operations per sample. The local spline interpolation of the waymark impulse response requires $3I$ MAC operations per waymark interval T_w for computing the spline coefficients according to (3.15) and $4I$ MAC operations per sample for interpolating the impulse response at the rate f_s according to (3.13). Note that the B-spline (3.14) does not need to be computed in real time as it can be precomputed on a uniform grid within the signal sampling interval T; e.g., the precomputation with a grid step of $T/1000$ would only require $4 \cdot 10^3$ memory, while introducing a negligible error in the recovered signal [50]. Similarly, local spline interpolation of the signal according to the composite delay requires 4 MAC operations per sample. In total, as $T_w \gg T$, the part indicated as 'Software/Hardware Implementation' in Fig.3.2 requires approximately $5I + 4$ MAC operations per sample, or about $5If_s$ MACs/s. As

Table 3.2: Channel simulator

Input:

$c(z)$, ρ, $[r(t), z(t)]$, $[\omega^-, \omega^+]$, T, $\tau_{\min}$, $\tau_{\max}$, $\delta\omega$, T_w, I, N

Initialization:

Repeat for $m = -2, \ldots, 3$**:**

Compute the frequency response $p_m(\omega_k)$, $k = 0, \ldots, K-1$,

for position $[r(mT_w), z(mT_w)]$ using acoustic field computation

Find the composite delay τ_m using a delay adjustment algorithm

Compute $\tilde{p}_m(\omega_k) = p_m(\omega_k) \exp(j\omega_k \tau_m)$

Compute the *waymark* impulse response $\tilde{\mathbf{h}}_m$ with elements:

$\tilde{h}_m(iT) = \Re\left\{ \frac{2}{K} \sum_{k=0}^{K-1} \tilde{p}_m(\omega_k) \exp(j\omega_k iT) \right\}$, $i = 0, \ldots, I-1$

Repeat for $m = -1, \ldots, 2$**:**

Compute spline coefficients for the impulse response:

$\mathbf{c}_m = a_{-1}\tilde{\mathbf{h}}_{m-1} + a_0\tilde{\mathbf{h}}_m + a_1\tilde{\mathbf{h}}_{m+1}$

Compute a spline coefficient for the composite delay:

$\theta_m = a_{-1}\tau_{m-1} + a_0\tau_m + a_1\tau_{m+1}$

Form an $I \times 4$ matrix: $\mathbf{C} = [\mathbf{c}_{-1}\ \mathbf{c}_0\ \mathbf{c}_1\ \mathbf{c}_2\ ]$

Form an 4×1 vector: $\boldsymbol{\theta} = [\theta_{-1}\ \theta_0\ \theta_1\ \theta_2]^T$

Set $T_h = T_w$

Repeat for $n = 0 : N-1$**:**

if $nT > T_h$

$m = m + 1$, $T_h = T_h + T_w$

Compute the frequency response $p_m(\omega_k)$, $k = 0, \ldots, K-1$, for position $[r(mT_w), z(mT_w)]$

Find the composite delay τ_m

Compute $\tilde{p}_m(\omega_k) = p_m(\omega_k) \exp(j\omega_k \tau_m)$

Compute the *waymark* impulse response $\tilde{\mathbf{h}}_m$

Compute spline coefficients for the impulse response:

$\mathbf{c} = a_{-1}\tilde{\mathbf{h}}_{m-2} + a_0\tilde{\mathbf{h}}_{m-1} + a_1\tilde{\mathbf{h}}_m$

Update $\mathbf{C}$: $\mathbf{C}^{(u)} = \mathbf{C}^{(u+1)}$, $u = 1, 2, 3$, and $\mathbf{C}^{(4)} = \mathbf{c}$

Compute spline coefficient for the composite delay:

$\theta_m = a_{-1}\tau_{m-2} + a_0\tau_{m-1} + a_1\tau_m$

Update $\boldsymbol{\theta}$: $\boldsymbol{\theta} = [\theta_{m-3}\ \theta_{m-2}\ \theta_{m-1}\ \theta_m]^T$

end if

$\xi = nT - T_h + T_w$

$\mathbf{b}(nT) = [b(\xi + T_w)\ b(\xi)\ b(\xi - T_w)\ b(\xi - 2T_w)]^T$

Spline interpolation of the impulse response: $\hat{\mathbf{h}} = \mathbf{C}\mathbf{b}$

Spline interpolation of the composite delay: $\hat{\tau}(nT) = \boldsymbol{\theta}^T\mathbf{b}$

Interpolation of the signal: $\tilde{s}(iT) \leftarrow s(nT - iT - \hat{\tau}(nT))$

Convolution: $y(nT) = \sum_{i=0}^{I-1} \hat{h}(nT, iT)\tilde{s}(iT)$

the number of taps $I \approx \tau f_s$, where τ is the multipath delay spread, the number of operations has a quadratic dependence on the sampling frequency f_s. E.g., for the Pacific ocean experiment described in Section 3.4 with the sampling frequency $f_s = 12288$ Hz and the number of filter taps $I = 7000$, the complexity is $5 \times 12288 \times 7000 \approx 430 \cdot 10^6$ MACs/s. This can be implemented in real time using modern DSP or/and FPGA design platforms. Note that the complexity of this part of the simulator is almost independent of the waymark interval T_w, i.e. the speed of the source. However, complexity of the part dealing with computations at the waymark rate T_w will be proportionally increased.

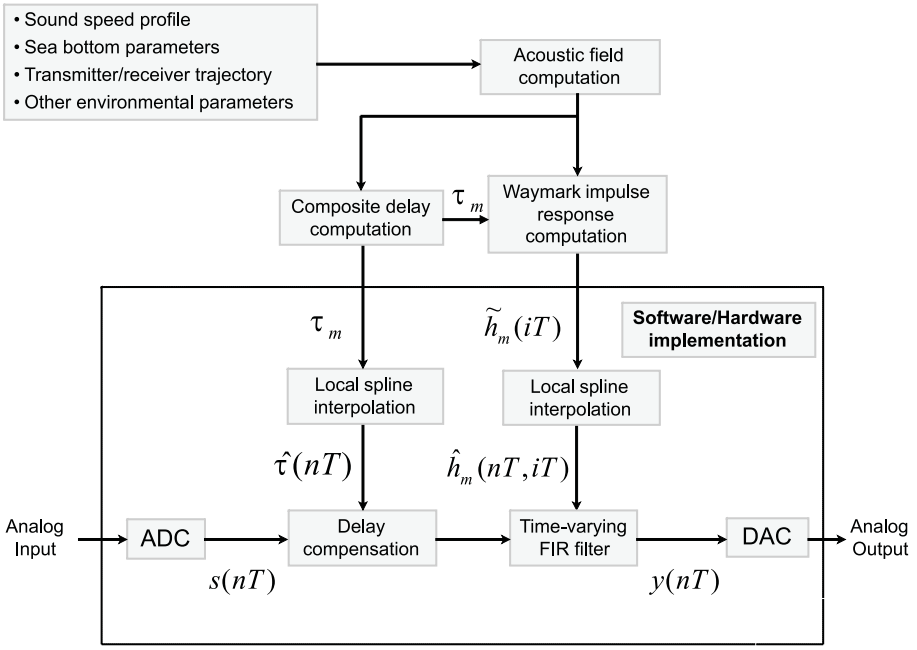


Figure 3.2: Underwater acoustic channel simulator.

3.3 Shallow water experiment

In this section, we apply the proposed approach to model signal transmission in an environment that corresponds to the shallow water experiment SWellEx-96 (Event S5) [60]. We show that the numerical simulation results match well to the experimental results.

During the experiment, a ship towing a deep source at a supposed depth of 54 m moved from a distance of 9 km towards and beyond a vertical line array (VLA) at a speed of 2.5 m/s. Our analysis is based on data collected on the VLA consisting of 21 hydrophones at

depths between 94.125 m and 212.25 m. The signal sampling rate is $f_s = 1/T = 1500$ Hz. The source transmitted simultaneously a set of 13 tones at frequencies from 49 Hz to 388 Hz. The SSP $c(z)$ from [60] shown in Fig. 3.3 is used in our simulation. In [61] [62],

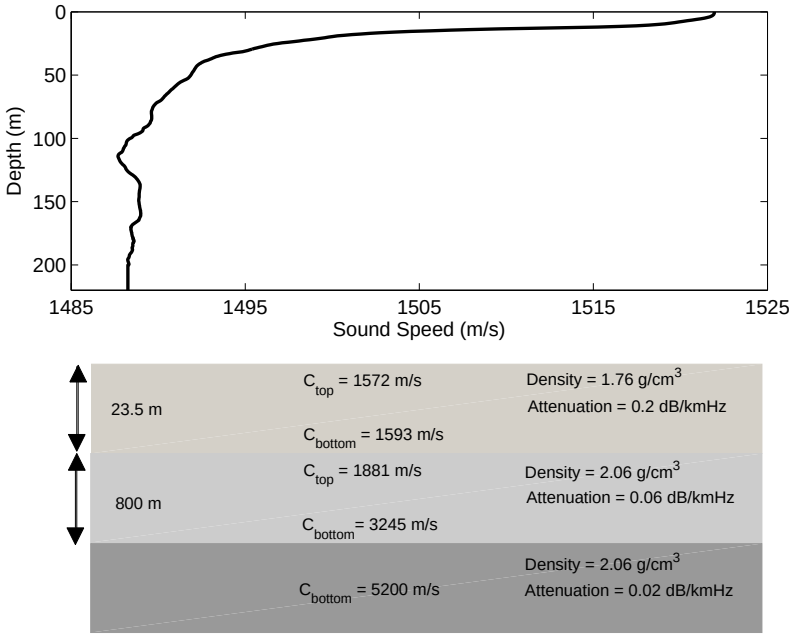


Figure 3.3: SSP for the experiment SWellEx-96 (Event S5).

we estimated the source range trajectory for this experiment. The estimate well matches to GPS measurements as shown in Fig.3.4(a). An estimate of the depth trajectory is shown in Fig.3.4(b). The estimated trajectory is used in our simulation to model the signals received at the 21 hydrophones.

To determine the waymark sampling period, we use the approach described in Subsection 3.2.5 as follows. E.g., for the distance $r = 4.5$ km, the multipath delay spread found from the field computation is approximately $\tau = 0.3$ s. We need to choose the range sampling interval Δr so that $\Delta r \ll (T/\tau)r$. As $T = 1/1500$ s, we have $\Delta r \ll 10$ m. With the speed of the transmitter 2.5 m/s, we obtain that the waymark sampling interval should satisfy $T_w \ll 4$ s; in our simulation, we use $T_w = 0.5$ s.

According to Fig.3.4(a), the maximum distance between the transmitter and VLA is about 9 km, which results in a 6 s propagation time. Without the delay adjustment, the frequency step $\delta\omega$ in computing the acoustic field should be set to a value $\delta\omega/(2\pi) <$

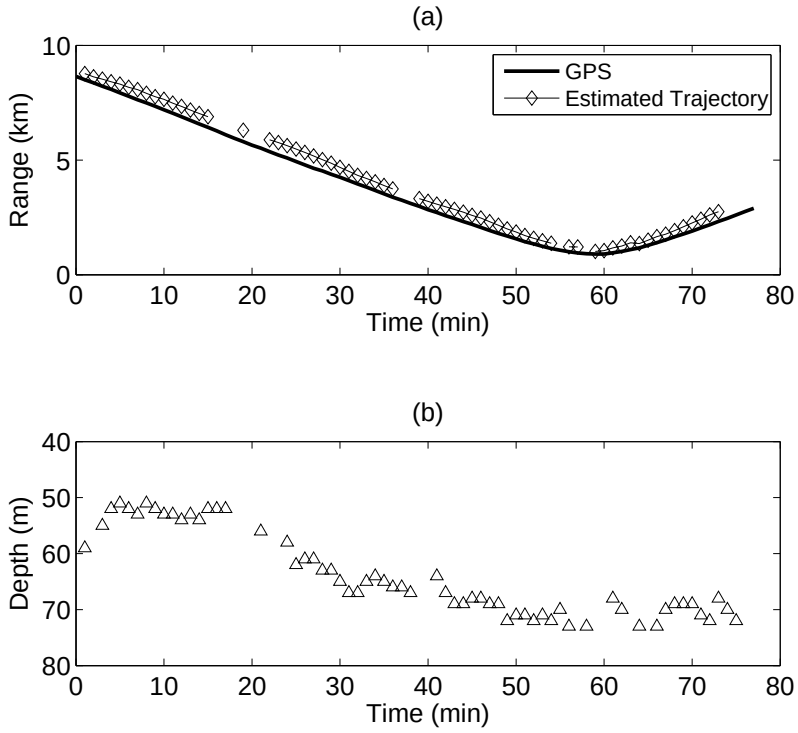


Figure 3.4: Trajectory of the source in the experiment SWellEx-96 (Event S5): (a) estimated range and range from GPS measurements; (b) estimated depth.

0.08 Hz. With the delay adjustment, since the maximum multipath delay spread is 0.3 s, the frequency step can be increased so that $\delta\omega/(2\pi) < 1.5$ Hz. In our simulation, the channel frequency response is computed with a frequency step $\delta\omega/(2\pi) = 1$ Hz in a frequency interval $[40, 410]$ Hz. The normal mode program KRAKEN [1] is used for the field computation.

Firstly, the effectiveness of the delay adjustment is examined. We compute the cross correlation between impulse responses found from the field computation at different positions. The correlation as a function of delay α between two positions is calculated as

$$\nu(\alpha) = \frac{\mathbf{h}^T(t_0)\mathbf{h}(t_0+\alpha)}{\sqrt{\mathbf{h}^T(t_0)\mathbf{h}(t_0)}\sqrt{\mathbf{h}^T(t_0+\alpha)\mathbf{h}(t_0+\alpha)}}, \tag{3.18}$$

where $\mathbf{h}(t_0)$ is the channel impulse response at a distance $r(t_0) \in [4, 5]$ km. Fig.3.5 shows the correlation averaged over the range interval $[4, 5]$ km versus the distance corresponding to α with and without the delay adjustment. Without the delay adjustment, the correlation rapidly decays as the distance increases; for $T_w = 0.5$ s (i.e. for a distance of

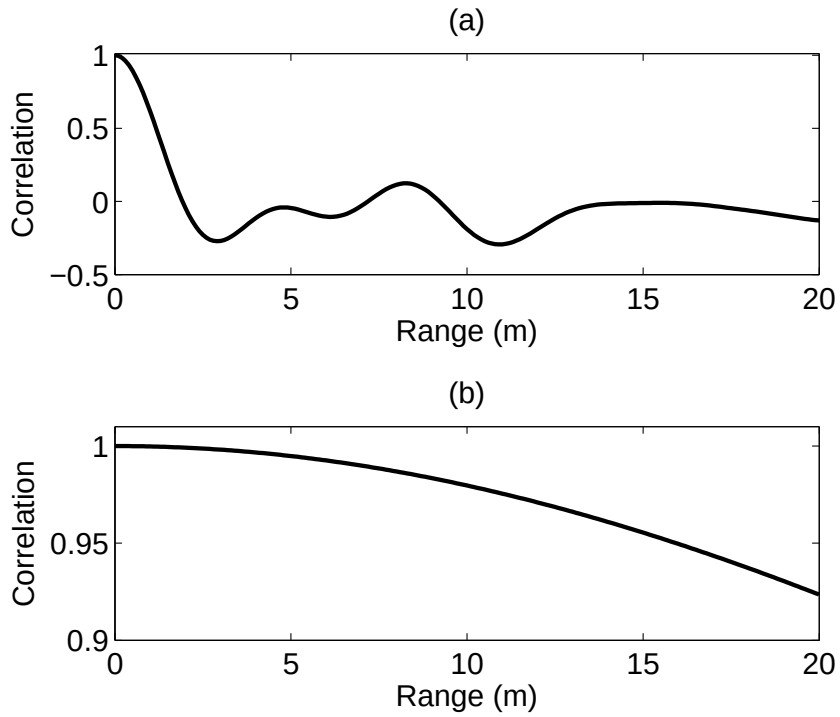


Figure 3.5: Correlation of impulse response versus distance for the experiment SWellEx-96: (a) without delay adjustment; (b) with delay adjustment.

1.25 m), the correlation is as low as $\nu = 0.45$. After the delay adjustment, the correlation is increased to $\nu = 0.9997$. Fig.3.6 shows the MSE between the impulse response obtained by spline approximation and the original (from the field computation) channel impulse response in the range interval $[4, 5]$ km. The MSE is averaged over each waymark sampling period, where the MSE is calculated as:

$$\text{MSE} = \frac{||\hat{\mathbf{h}}(t) - \mathbf{h}(t)||_2^2}{||\mathbf{h}(t)||_2^2}. \tag{3.19}$$

It is seen that the MSE is below -68 dB, which shows that here the spline interpolation is very accurate.

Fig.3.7 compares the Doppler spread of the tones obtained from the experimental and simulated received signals. The Doppler spectra are obtained by averaging periodograms of the received tones over 100s-long snapshots and over the 21 receiver hydrophones. The frequency shifts of the tones due to the constant speed of the ship and geometry of the experiment are removed as described in [61] [62]. It is seen that the Doppler spread of the modeled signals match well to that obtained from the experimental data. The mismatch

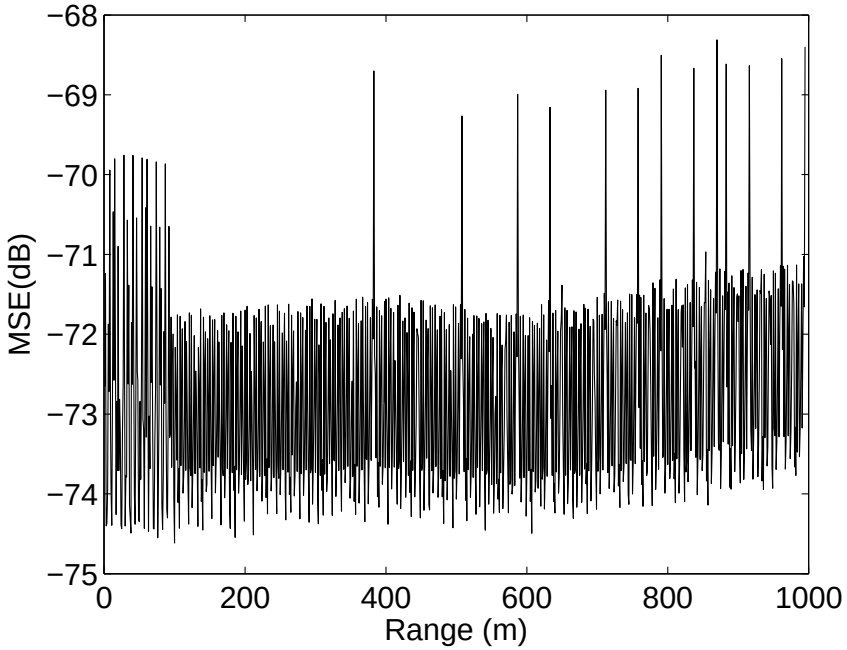


Figure 3.6: MSE between the approximated and original channel impulse response for the experiment SWellEx-96.

in the floor level is due to the noise present in the experimental data and not added into the simulated signals.

3.4 Deep water experiment

In this section, we verify the proposed channel model on data obtained in a deep water experiment in the Pacific Ocean [44]. During the experiment, an acoustic source was towed by a ship moving towards the receiver at a supposed speed of 6 m/s and depth of 200 m. An omnidirectional hydrophone, positioned at a nominal depth of 400 m, was used to record the received signal. The distance between the transmitter and receiver varied from 42 km to 40 km. In the vertical plane, the transducer beampattern is approximated as having a 0dB-gain for grazing angles $|\alpha| \leq 15^\circ$ and -10dB for $15^\circ < |\alpha| < 25^\circ$. SSPs measured in the experiment are shown in Fig.3.8. The SSP averaged over the measured SSPs is used in the simulation. The ray tracing program BELLHOP [59] is used to generate eigenpaths.

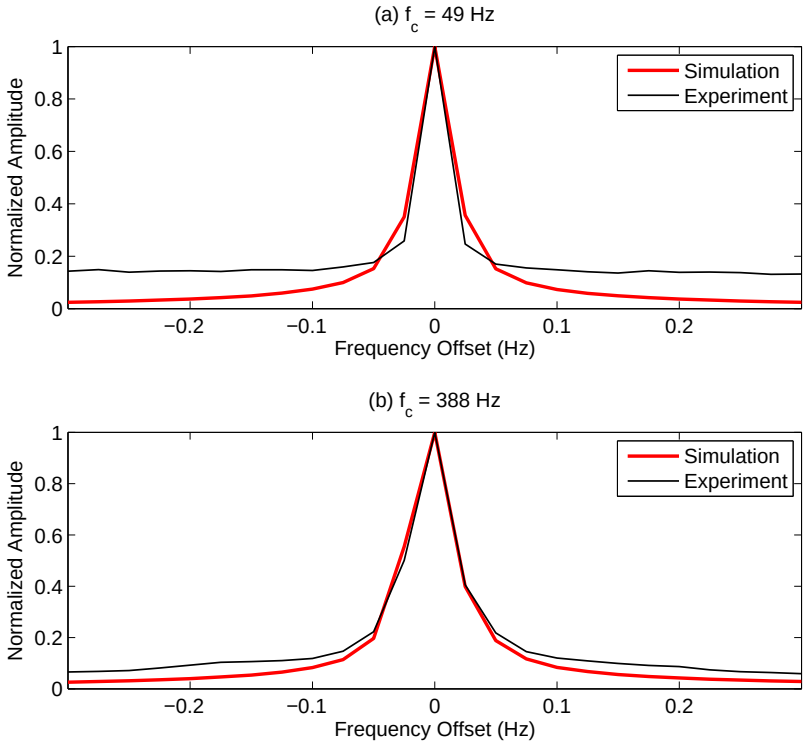


Figure 3.7: Doppler spread of two tones (of frequencies f_c) obtained from the experimental and simulated signals.

In the experiment, 340 coded OFDM symbols were transmitted. One OFDM symbol is given by [44, 63]

$$s(t) = \sum_{k=0}^{N_{\text{sc}}-1} \cos[2\pi f_k + \phi(k)], \tag{3.20}$$

where the number of subcarriers $N_{\text{sc}} = 1024$, $f_k = f_c - F/2 + k/T_s$, $f_c = 3072$ Hz, the frequency bandwidth of the signal is $F = N_{\text{sc}}/T_{\text{OFDM}} = 1024$ Hz, and $T_{\text{OFDM}} = 1$ s is the OFDM symbol duration. The phase modulation $\phi(k)$ of the subcarriers is given by

$$\sqrt{2}e^{j\phi(k)} = M_2(k) + jM_1(k), \tag{3.21}$$

where $M_2(k)$ is a coded bit transmitted at the kth subcarrier, and $M_1(k)$ is a pilot bit; the sequences $M_1(k)$ and $M_2(k)$ are binary with values ± 1. The sequence $M_2(k)$ is a rate $1/2$ convolutional code generated by polynomials $[247, 371]$ in octal [64]. The OFDM symbols are transmitted one-by-one without any guard interval. Thus, the information data rate in the experiment is about 500 bit/s, or equivalently ≈ 0.5 bit/s/Hz. The absence of the guard interval and the periodical transmitted pilot allows a simple procedure of time

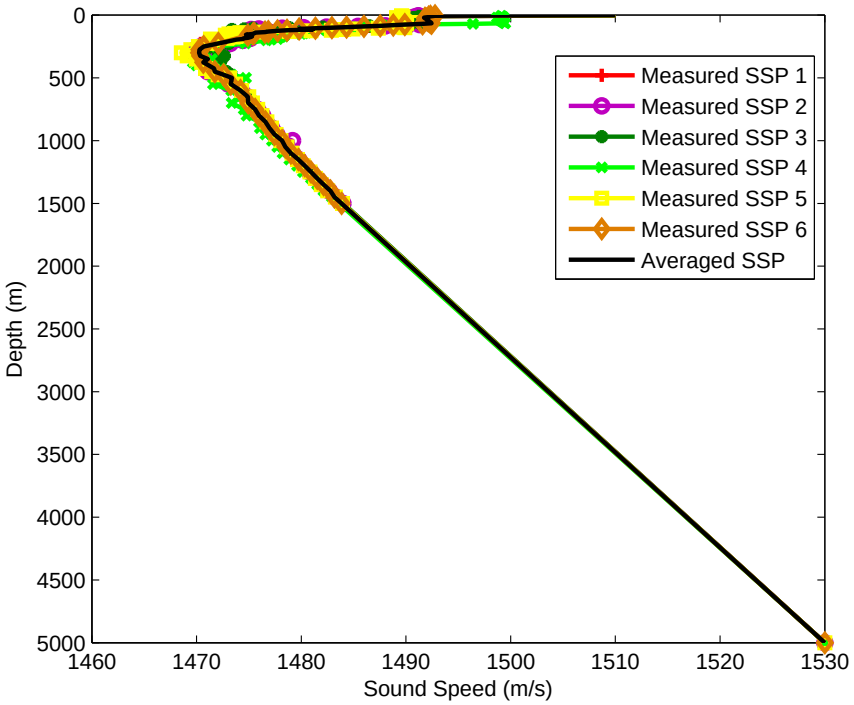


Figure 3.8: SSPs $c(z)$ for the Pacific Ocean experiment.

synchronization. Moreover, since the channel is doubly-selective, there is no significant benefit in using a guard interval, which can also reduce the transmission efficiency.

The receiver is briefly described as follows.

Firstly, a time-varying Doppler compression factor is estimated with a time step $T_{\text{est}} \leq T_{\text{OFDM}}$. This is done by computing the cross-ambiguity function between the received and pilot signals on a 2D-grid of delay and compression factor and finding the maximum on the 2D-grid [29, 44]. The compression factor estimate is further rectified by using parabolic interpolation as described in [65]. The compression factor estimates are linearly interpolated to the signal sampling rate and used to compensate for the time-varying Doppler effect by resampling the signal with the time-varying compression factor. The signal is then transformed into a complex-envelope signal within the frequency range $[-512, +512]$ Hz.

Secondly, linear time-domain FIR equalization with equalizer coefficients derived from channel estimates is applied to the complex-envelope signal. The main purpose

of the equalizer is to reduce the multipath delay spread. The channel estimation is based on computing cross-correlation between the pilot and complex-envelope signals and is performed with the same time step T_{est} as the estimation of the Doppler compression factor. The equalized signal is then transformed into the frequency domain using the Fast Fourier Transform.

Finally, N_{turbo} turbo iterations ($N_{\text{turbo}} = 3$ in our case) are repeated, each performing frequency-domain channel estimation, equalization and soft-decision Viterbi decoding [64]. The channel estimation is based on the BEM with complex exponentials. For the channel estimation, the pilot and tentative estimates of information symbols (zeros at the first turbo iteration) produced by the Viterbi decoder are used.

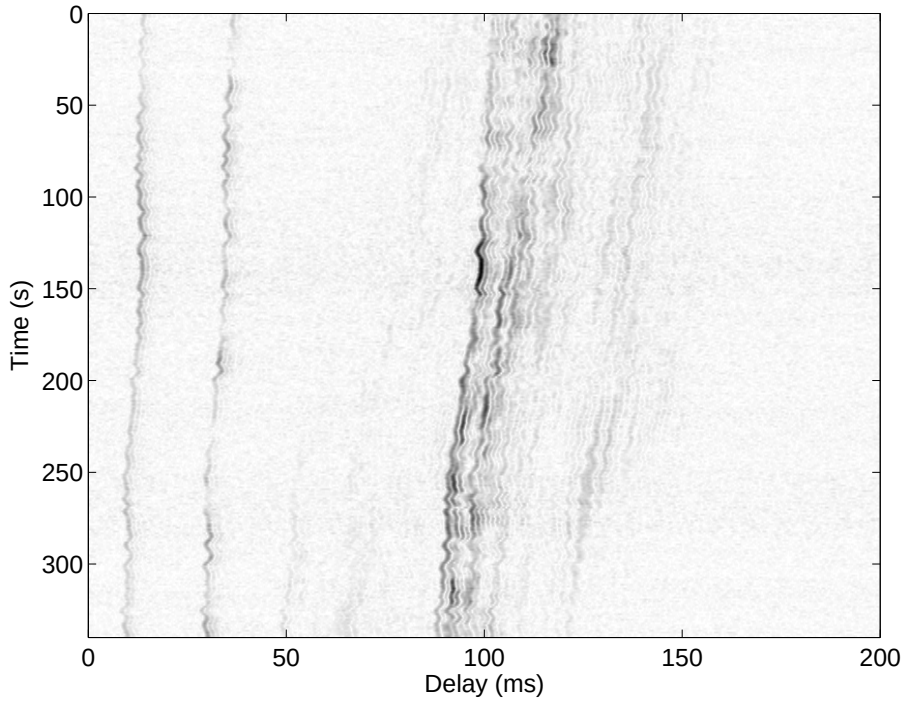
Fig.3.9(a) shows the impulse response estimates obtained from the experimental data. There can be seen periodic fluctuations of multipath delays. Analysis of the fluctuations shows that they are synchronous, i.e. all delays are increased or decreased simultaneously. This is an indication of range fluctuations as opposed to depth fluctuations of the source, in which case, if delays of multipaths arriving at the receiver at positive grating angles increased, delays of multipaths arriving at negative angles would decrease and vice versa. In the case of depth fluctuations, we would see in Fig.3.9(a) opposite delay fluctuations for some multipath components. The range fluctuations are caused by the towing of the transducer by the surface ship. This can be explained by the drawing in Fig.3.10. When towing the transducer, due to impact of the surface waves on the ship, the tether was randomly sweeping in angle, which resulted in the range fluctuations with an average amplitude of 1 m. The spectrum of the fluctuations matches well to the surface waves with an average period of about 10 s.

In the simulation, to match the experimental data, the source is modeled as moving at a constant depth with a time-varying speed (with range fluctuations)

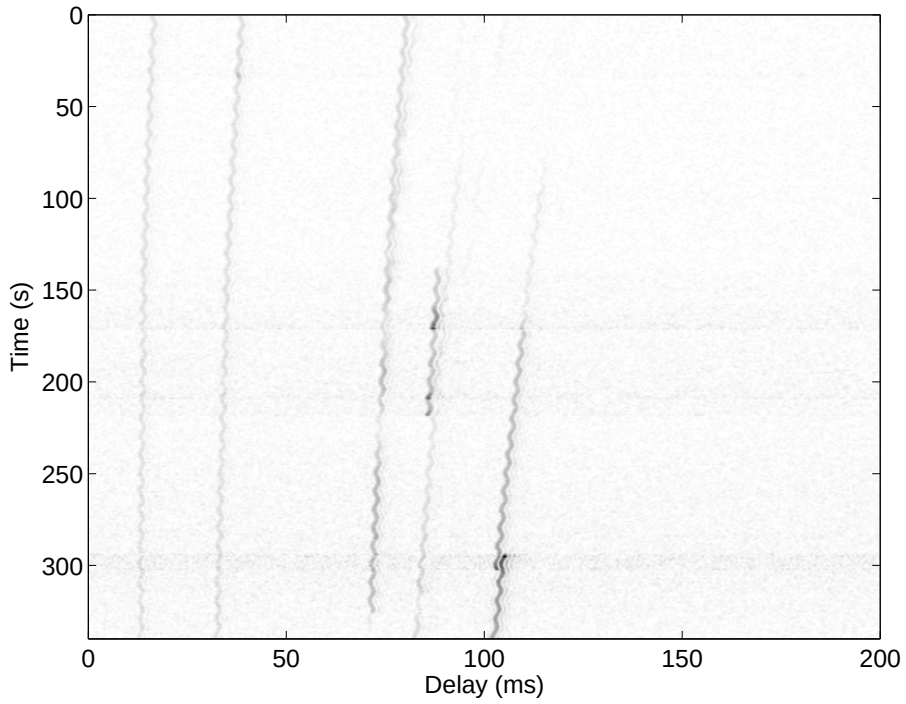
$$r(t) = r_o - v_o t + r_{\text{fluct}}(t), \tag{3.22}$$

where $r_0 = 42.16$ km and $v_0 = 6$ m/s. In the first part of the simulation, we use a simplified *'sinusoidal movement'*: $r_{\text{fluct}}(t) = \mu \sin(2\pi t/T)$ with $\mu = 1$ m. In the second part of the simulation, we use $r_{\text{fluct}}(t)$ derived from the experimental data.

To choose the waymark sampling interval, we apply the approach described in Sub-



(a) From experimental data



(b) From simulated data for the sinusoid movement

Figure 3.9: Estimates of the channel impulse response.

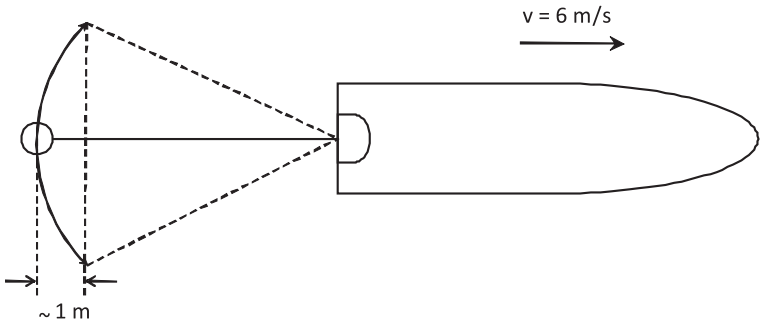


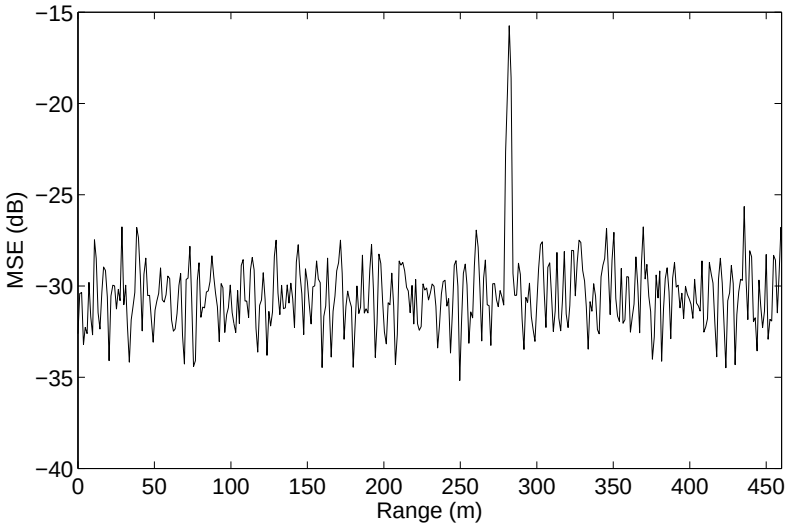
Figure 3.10: Tether angle sweeping when towing the transducer.

section 3.2.5. The range step Δr should be chosen to satisfy $\Delta r \ll (T/\tau)r$. In the experiment, we have $r \approx 40$ km, $T = 6.9\ 10^{-5}$ s (the signal sampling rate is $4 \cdot 3072 = 12288$ Hz), and the multipath delay spread $\tau < 0.5$ s (in addition to multipaths shown in Fig.3.9(a) there are weak multipaths with delays close to 0.5 s). Thus, we obtain that $\Delta r \ll 5.5$ m. With the speed of the transmitter 6 m/s, we obtain that the waymark sampling interval should satisfy $T_w \ll 0.9$ s. We consider two cases: $T_w = 0.2$ s and $T_w = 0.1$ s. The frequency step for computing the channel frequency response is set to $\delta\omega/(2\pi) = 1$ Hz.

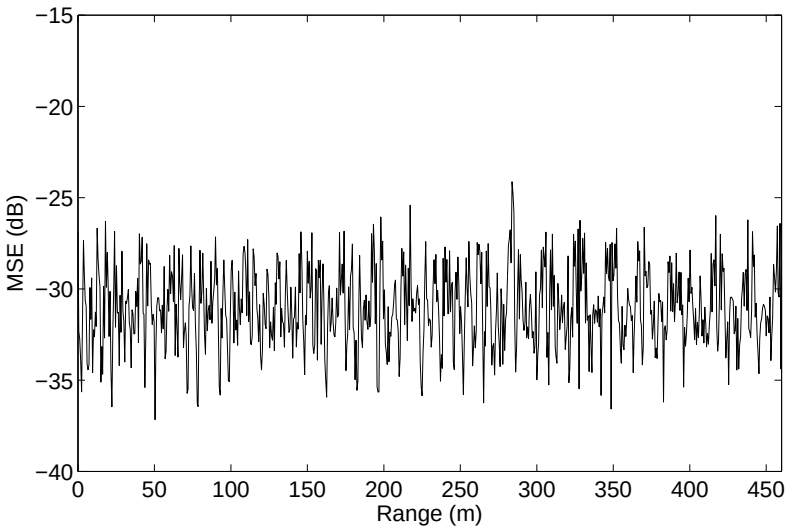
Fig.3.11 shows the MSE between the impulse response obtained from the acoustic field computation and the result of the spline interpolation for the first 450 m of the trajectory. For $T_w = 0.2$ s, the MSE is close to -30 dB; however, at the distance 280 m, the MSE increases to a level of -16 dB. The MSE peak is caused by change in the multipath structure; specifically, at this position some multipath components disappear. When reducing the waymark sampling interval to $T_w = 0.1$ s, the MSE peak is reduced to the level -24 dB.

We now model the signal transmission for the sinusoidal movement and estimate the time-varying impulse response from the simulated signal; the impulse response is shown in Fig.3.9(b). Comparing Fig.3.9(a) and (b), it is seen that the proposed simulator accurately describes the effect caused by fluctuations of the source movement.

We now model the source movement with $r_{\text{fluct}}(t)$ derived from the experimental data. Specifically, the time varying speed is computed using the time-varying Doppler compression factor estimated from the experimental data. Fig.3.12(a) shows the compression factor variations for the strongest multipath component estimated from the experimen-

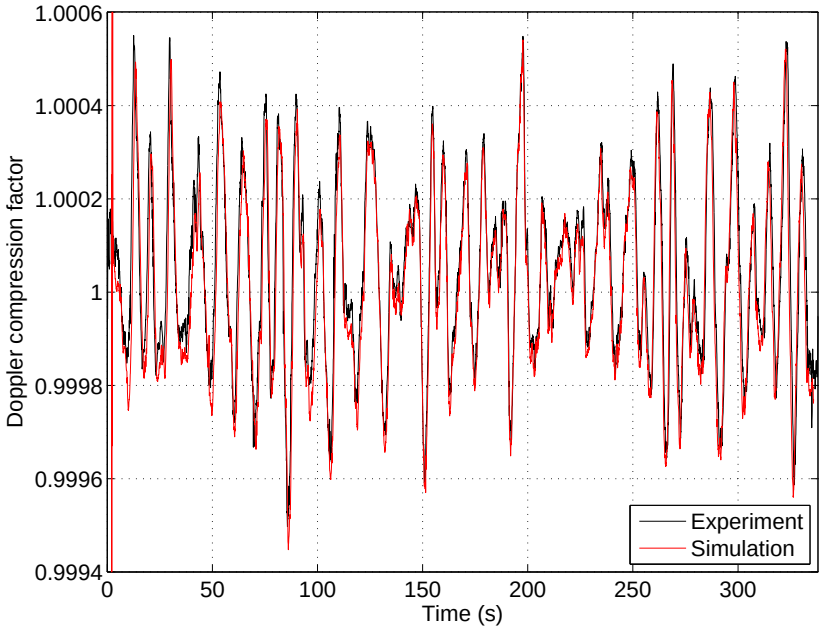


(a) $T_w = 0.2$s

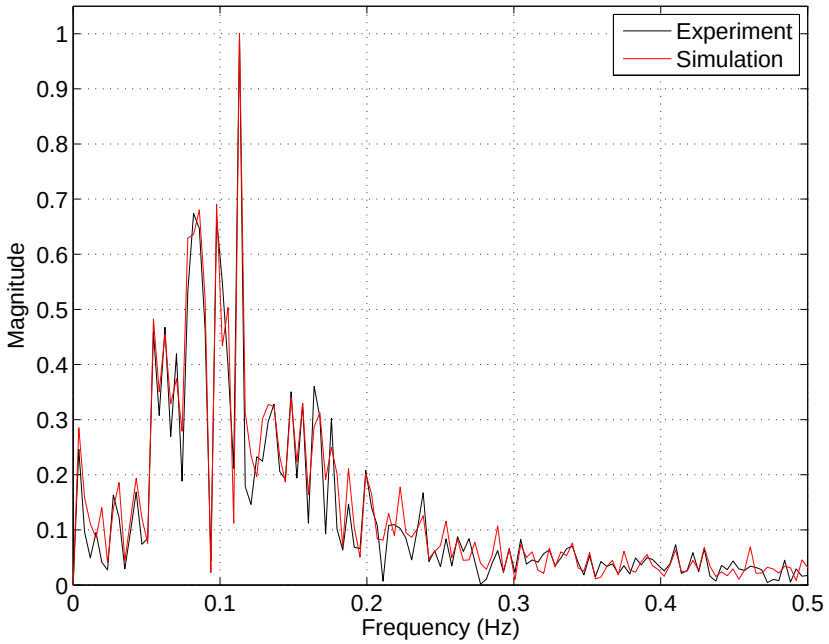


(b) $T_w = 0.1$s

Figure 3.11: MSE between the impulse response obtained from the acoustic field computation and the impulse response obtained by the local spline interpolation.



(a) Fluctuations of the compression factor in time



(b) Spectrum of compression factor fluctuations

Figure 3.12: Fluctuations of the compression factor in the experimental and simulated signals.

tal and simulated signals. A constant compression factor due to the constant speed v_0 is removed to emphasize the fluctuations. Fig.3.12(b) shows the spectrum of the fluctuations. It is seen that the results obtained in simulation match very well to that from the experiment.

Next, the Bit Error Rate (BER) and MSE performance of the receiver is investigated. In Fig.3.13, the BER is shown as a function of the estimation interval T_{est}. The average signal to noise ratio (SNR) in the simulated signals is set to 25 dB by adding white noise; this is the same SNR as measured in the experimental data. In addition to the two scenarios where the trajectory has range fluctuations ($r_{\text{fluct}}(t) \neq 0$), we have also simulated data for the scenario with no range fluctuations ($r_{\text{fluct}}(t) = 0$), i.e. when the source is moving with a constant speed of 6 m/s. By comparing the BER performance of the receiver in the three simulation scenarios and in the experiment, we can conclude the following. The simulation results for the sinusoidal and real movements that take into account the range fluctuations both match well to the experimental data. However, the simulation results for the scenario with no range fluctuations show a lower BER. This is well supported by the Doppler spectrum for this scenario in Fig.3.15, which shows slower channel variations compared to the variations in the case of range fluctuations (see Fig.3.12). It is seen that the threshold interval T_{est} beyond which the BER significantly increases is now higher and the floor BER level is lower than in the other cases. Fig.3.14 shows the MSE of the soft demodulation output in the receiver. Comparing it with results in Fig.3.13, it is seen that the MSE exhibits a behavior similar to that of the BER performance. Thus, taking into account the range fluctuations in the trajectory is useful for accurate prediction of the receiver performance. Simulation of the real movement is possible if the data like those shown in Fig.3.12(a) are available. If not, the sinusoidal movement provides a good approximation to the scenario with the real movement.

3.5 Summary

In this chapter, we have developed a method for modeling underwater acoustic signal transmission for moving transmitter and/or receiver, particularly for underwater acoustic communications. The proposed method is based on approximation of the time-varying

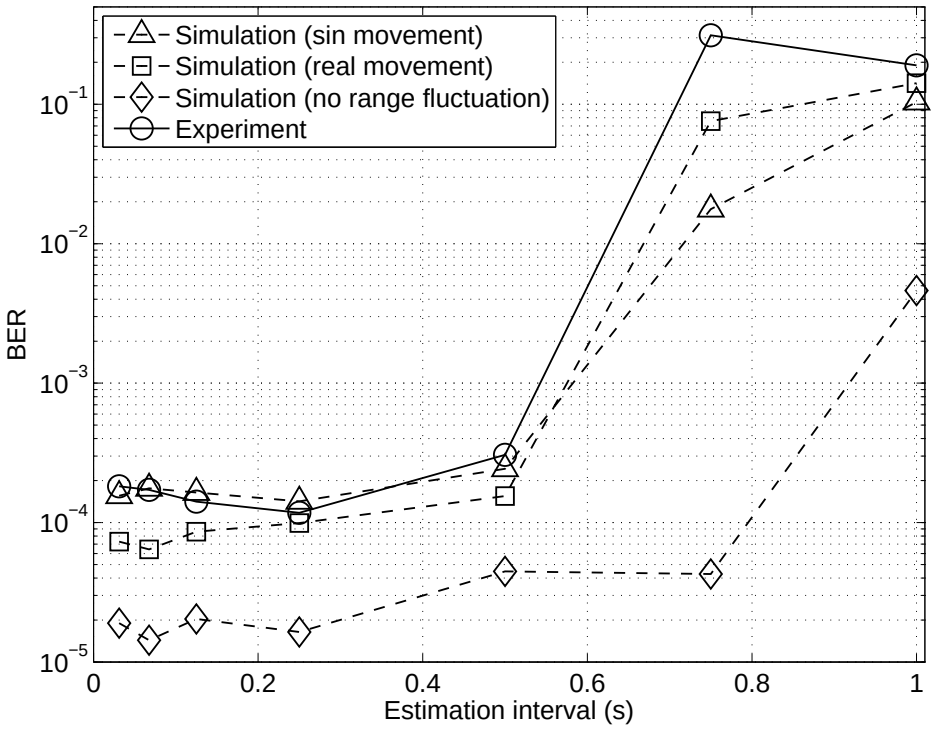


Figure 3.13: BER versus the Doppler and channel estimation interval T_{est}.

channel impulse response along the transmitter/receiver trajectory. This is implemented by sampling the trajectory and using local splines for the approximation. The proposed method has a low complexity and low memory consumption. Besides, it can be implemented in a recursive form and, thus, can be used for developing real-time simulators for long-duration communication sessions. The proposed method has been verified by comparing the simulated data with data from real ocean experiments in shallow and deep water. For the shallow water experiment, we have shown that the Doppler spectrum of the tones transmitted by a moving source is similar in the simulation and experiment. For the deep water experiment, we have investigated the influence of the channel estimation interval on the detection performance of an OFDM receiver. It has been shown that the receiver performance is similar in the simulation and experiment.

However, it is worth noting that the two experiments used to verify the channel model exploited low-frequency signals: $49-388$ Hz in the shallow water and $2560-3584$ Hz in the deep water. Therefore, the verification using experiments with signal transmission at higher frequencies is highly desirable. Also, in the experiments used for the verification, scattering from the see surface was not essential for the acoustic propagation. However,

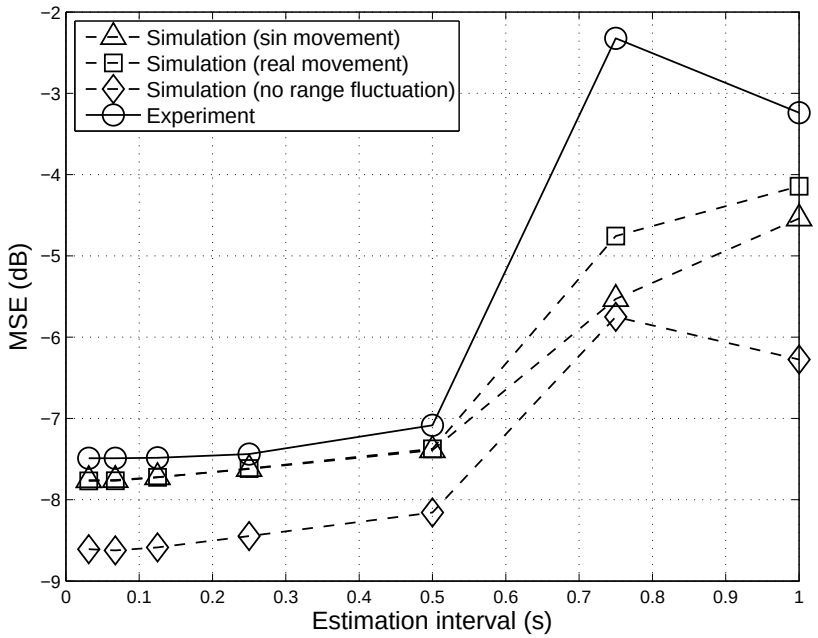


Figure 3.14: MSE versus the Doppler and channel estimation interval T_{est}.

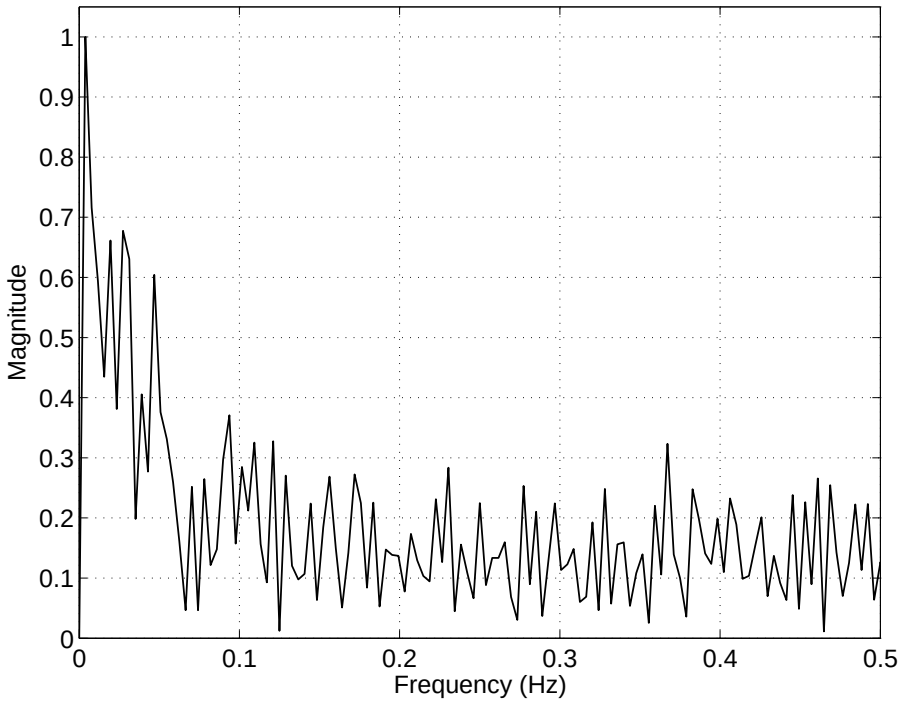


Figure 3.15: Doppler spectrum of compression factor fluctuations obtained from the simulation: the scenario with no range fluctuation.

in many practical situations, the surface has significant impact on the propagated signal [39, 40, 66–68]. Thus, the proposed method can be used for modeling transmission of low-frequency acoustic signals for underwater communications and other systems, where the source and/or receiver are moving, and in channels, where the surface scattering is not significant. Verifying the channel model using higher frequency sea experiments and incorporating in the model effects of acoustic scattering from the sea surface are the subjects of our future research.

Chapter 4

Underwater Localization of Multiple Sources using Basis Pursuit De-Noising

Contents

4.1 Introduction

Antenna array beamforming based on the MF processing is an effective method for estimating positions of underwater acoustic sources [14, 15]. In the MF beamforming, for each hypothesized position, the acoustic propagation is modeled using known positions of the antenna elements and environmental parameters such as the SSP and bottom composition. As a result, a steering vector is assigned to each potential position. A correlation between the steering vectors for all the potential positions and the real signal is evaluated to form an ambiguity surface [69]. Estimates of the source positions are then obtained according to highest peaks of the ambiguity surface, e.g, in case of locating two

sources, the source positions are determined by the highest and the second highest peak of the ambiguity surface. Previously, many MF techniques have been proposed that show good performance for single source localization in simulated scenarios and real ocean experiments (e.g. see [14, 61, 62, 70] and references therein). However, few MF techniques [14, 15, 62, 71] can provide good performance in detecting positions of multiple sources simultaneously transmitting signals at the same frequencies, especially when the sources are close to each other.

In the underwater localization, the number of potential positions is often significantly larger than the number of sources. This allows formulation of the localization problem in terms of sparse representation of source positions. The problem can then be reformulated as finding the sparsest representation of the source positions, and for solving it greedy methods [72] or l_1 norm minimization basis pursuit de-noising (BPDN) [73] can be used. In this chapter, we consider the l_1 norm minimization problem which yields accurate solutions in certain scenarios where greedy methods fail [73] [74]. Specifically, we deal with the BPDN problem [75] [76].

Classical methods for solving the BPDN problem adopt convex optimization to obtain a sparse solution (see [77], [78], [79] and references therein). The complexity of these methods can be high when the size of the BPDN problem, i.e. the number of potential source positions, is large. An alternative method with lower complexity is the Homotopy method [73] [80].

In this chapter, we explore the idea of the Homotopy method and propose a technique that is well suited to the MF localization. The main contribution of this chapter is as follows. We formulate the underwater acoustic source localization as the complex-valued BPDN problem and show high efficiency of such an approach. The BPDN problem is solved for multi-frequency scenarios to benefit from the frequency diversity. The Homotopy approach combined with the CD search [81, 82] is used for solving this multi-frequency BPDN problem. Grid refinement is further used to reduce the complexity of solving the BPDN problem. Comprehensive simulation is carried out to evaluate the performance of the proposed method in resolving sources close to each other, locating multiple sources and robustness to noise and interference. The effectiveness of the proposed method is verified using real ocean experimental data.

The rest of this chapter is organized as follows. In Section 4.2, the signal model is described and the source localization problem is formulated; the conventional MF and Iterative Adaptive Algorithm (IAA) [71] MF processing techniques are presented. Section 4.3 describes the proposed technique; the CD search and grid refinement method are also presented here. In Section 4.4, the proposed method is evaluated by applying it to simulated and real experimental data. Finally, Section 4.5 gives conclusions.

4.2 Signal model for source localization and matched field processing

In a scenario with K potential source positions, the signal at frequency f_m, $m = 1, \ldots, M$, received by an N-element antenna array can be modeled as

$$\mathbf{y}(f_m) = \sum_{k=1}^{K} s_k(f_m)\mathbf{p}_k(f_m) + \mathbf{e}(f_m) \tag{4.1}$$

where $\mathbf{p}_k(f_m)$ is a $N \times 1$ steering vector corresponding to the kth potential position, and $\mathbf{e}(f_m)$ is the noise at the receiver. With the MF processing, the steering vector $\mathbf{p}_k(f_m)$ is computed using an acoustic propagation model with known environmental parameters such as the SSP and bottom composition and known positions of the antenna elements.

For further derivation, it is convenient to rewrite the signal model by introducing the following notations:

$$\begin{aligned} \mathbf{a}_k(f_m) &= \mathbf{p}_k(f_m)/||\mathbf{p}_k(f_m)||_2 \\ x_k(f_m) &= s_k(f_m)||\mathbf{p}_k(f_m)||_2. \end{aligned}$$

Then (4.1) can be rewritten as

$$\mathbf{y}(f_m) = \mathbf{A}(f_m)\mathbf{x}(f_m) + \mathbf{e}(f_m), \tag{4.2}$$

where $\mathbf{A}(f_m) = [\mathbf{a}_1(f_m), \mathbf{a}_2(f_m), \ldots, \mathbf{a}_K(f_m)]$ is the $N \times K$ steering matrix defining all the potential positions, $\mathbf{a}_k^H\mathbf{a}_k = 1$, and $\mathbf{x}(f_m) = [x_1(f_m), x_2(f_m), \ldots, x_K(f_m)]^T$ is a signal vector. The kth element of $\mathbf{x}(f_m)$ is considered to be a complex-valued signal generated by a source at the kth position at frequency f_m. The number of sources is

assumed to be much smaller than K, i.e. the vector $\mathbf{x}(f_m)$ is sparse. Based on this signal model, the localization can be formulated as finding non-zero elements in $\mathbf{x}(f_m)$, i.e. solving a sparse representation problem. For instance, searching sources in the range interval [0, 5] km with a resolution of 10 m and in the depth interval [0, 100] m with a resolution of 5 m, results in $K \approx 10^4$. If there are a few acoustic sources, we clearly deal with a sparse representation problem.

Many MF processors provide good performance in locating a single source, e.g. the coherent matched phase processor [16] or the minimum variance distortionless filter MF processor [19]. However, they are not efficient in multiple source scenarios. This is especially true when the number of snapshots is limited. An MF processor, which can deal with the multi-source localization problem, is the conventional (Bartlett) MF processor using the ambiguity function [14] [15]:

$$B_{\text{conv}}(\psi_k, f_m) = |\mathbf{a}_k^H(f_m)\mathbf{y}(f_m)|^2, \tag{4.3}$$

where $\psi_k = (r_k, d_k)$ is the kth potential source position, r_k is the horizontal distance between the source and receiver and d_k is the source depth. For multi-frequency scenarios, the ambiguity function can be computed by incoherently combining the M single frequency ambiguity functions [18]

$$\begin{aligned} B_{\text{conv}}(\psi_k) &= \frac{1}{M}\sum_{m=1}^{M} B_{\text{conv}}(\psi_k, f_m) \\ &= \frac{1}{M}\sum_{m=1}^{M} |\mathbf{a}_k^H(f_m)\mathbf{y}(f_m)|^2. \end{aligned} \tag{4.4}$$

Another MF processor, exploiting the Iterative Adaptive Algorithm (IAA) [71], is based on sparse representation and can provide better localization than the conventional MF processor [62]. The ambiguity surface in the IAA MF processor is given by

$$B_{\text{IAA}}(\psi_k, f_m) = |\hat{x}_k(f_m)|^2 . \tag{4.5}$$

The estimate $\hat{x}_k(f_m)$ of the source signal $x_k(f_m)$ at position ψ_k in an iteration is given by [71]

$$\hat{x}_k(f_m) = \frac{\mathbf{a}_k^H(f_m)\mathbf{R}^{-1}(f_m)\mathbf{y}(f_m)}{\mathbf{a}_k^H(f_m)\mathbf{R}^{-1}(f_m)\mathbf{a}_k(f_m)} \tag{4.6}$$

where $\mathbf{R}(f_m) = \mathbf{A}(f_m)\mathbf{P}(f_m)\mathbf{A}^H(f_m)$ and $\mathbf{P}(f_m)$ is a $K \times K$ diagonal matrix with diagonal entries $|\hat{x}_k(f_m)|^2$ obtained at the previous iteration. The ambiguity function for

multi-frequency signals can be obtained by incoherently combining the M single frequency ambiguity functions

$$\begin{aligned} B_{\mathrm{IAA}}(\psi_k) &= \frac{1}{M}\sum_{m=1}^{M} B_{\mathrm{IAA}}(\psi_k, f_m) \\ &= \frac{1}{M}\sum_{m=1}^{M} \left|\hat{x}_k(f_m)\right|^2 . \end{aligned} \tag{4.7}$$

The conventional and IAA processors will be used in Section 4.4 for comparison with the proposed method.

4.3 Basis Pursuit De-noising

Underwater acoustic sources can be localized by finding non-zero elements in the vector $\mathbf{x}(f_m)$. In a noise free case, this is equivalent to finding the sparsest representation of $\mathbf{x}(f_m)$ under the constraint

$$\mathbf{y}(f_m) = \mathbf{A}(f_m)\mathbf{x}(f_m).$$

This problem can then be reformulated as the l_0 norm minimization problem [83]:

$$\min_{\hat{\mathbf{x}}(f_m)} ||\hat{\mathbf{x}}(f_m)||_0 \ \text{ s.t. } \ \mathbf{y}(f_m) = \mathbf{A}(f_m)\hat{\mathbf{x}}(f_m), \tag{4.8}$$

where $\hat{\mathbf{x}}(f_m)$ is the solution at a frequency f_m. However, this is a non-convex combinatorial problem which is NP hard [84]. The sparse solution can be found approximately by minimizing the l_1 norm [83] [84] [74]:

$$\min_{\hat{\mathbf{x}}(f_m)} ||\hat{\mathbf{x}}(f_m)||_1 \ \text{ s.t. } \ \mathbf{y}(f_m) = \mathbf{A}(f_m)\hat{\mathbf{x}}(f_m). \tag{4.9}$$

This is a convex problem that can be solved using known techniques, e.g. the linear programming [77].

For noisy data, it is unrealistic to find a solution with exact fit so that $\mathbf{y}(f_m) = \mathbf{A}(f_m)\mathbf{x}(f_m)$. To account for the noise, the linear fit constraint is relaxed, which yields the BPDN problem [75] [76]:

$$\min_{\hat{\mathbf{x}}(f_m)} ||\hat{\mathbf{x}}(f_m)||_1 \ \text{ s.t. } \ ||\mathbf{y}(f_m) - \mathbf{A}(f_m)\hat{\mathbf{x}}(f_m)||_2^2 \leq \sigma$$

where σ is a positive constant. For the convenience of adopting convex optimization methods, the BPDN problem is represented in the unconstrained formulation [73]:

$$\min_{\hat{\mathbf{x}}(f_m)} J[\hat{\mathbf{x}}(f_m)] \tag{4.10}$$

where

$$J[\hat{\mathbf{x}}(f_m)] = \frac{1}{2}\left|\left|\mathbf{y}(f_m) - \mathbf{A}(f_m)\hat{\mathbf{x}}(f_m)\right|\right|_2^2 + \tau\left|\left|\hat{\mathbf{x}}(f_m)\right|\right|_1 \tag{4.11}$$

and τ is a positive parameter that determines a balance between the l_1 and l_2 norms.

The accuracy and sparsity of the solution $\hat{\mathbf{x}}(f_m)$ is sensitive to the parameter τ. Accurate choice of τ would require knowledge of the noise level and sparsity of $\mathbf{x}(f_m)$ [80], which maybe difficult to obtain. In the literature, there is analysis of methods in choosing proper values of τ (see [85] and the references therein). An alternative method for dealing with little knowledge of noise and sparsity is to obtain solutions for different values of τ and choose the best one. The Homotopy approach [73,80] allows finding a sparse solution to the BPDN problem for a set of τ with a low complexity which is comparable to that of greedy algorithms [73]. With this approach, the BPDN problem is first solved for a large τ. Then τ is discretely reduced and, for each τ, the BPDN problem is solved using the previous solution as an initialization, until a stopping criterion is satisfied. The solution is updated by adding or removing non-zero elements to/from the vector $\hat{\mathbf{x}}(f_m)$. Reducing τ transfers the l_1 minimization problem for large τ to a l_2 minimization problem for small τ.

Here, we adopt this idea and propose a technique that differs from the Homotopy approach in [73] in that the parameter τ is reduced gradually with a predefined rate and that elements are not removed from the solution vector; the latter however insignificantly affects the accuracy [73]. For each τ, a solution is found using a CD line search.

4.3.1 Solving the complex-valued BPDN for a single-frequency case

We consider an iterative process where at each iteration a new non-zero element is added to the current solution $\hat{\mathbf{x}}(f_m)$ to the problem (4.11). Suppose that we have already n non-zero elements in the current solution vector $\hat{\mathbf{x}}(f_m)$ and denote the support (the index set of non-zero elements) as $\mathbf{I}_m(n) = [I_m(1), \ldots, I_m(n)]$; the complementary set is $\bar{\mathbf{I}}_m(n)$.

For the current solution $\hat{\mathbf{x}}(f_m)$, we define a vector $\mathbf{r}(f_m) = \mathbf{y}(f_m) - \mathbf{A}(f_m)\hat{\mathbf{x}}(f_m)$ and a residual vector

$$\mathbf{b}(f_m) = \mathbf{A}^H(f_m)\mathbf{r}(f_m).$$

The new non-zero element is only added if it reduces the cost function. Below, we show that for $t \in \bar{\mathbf{I}}_m(n)$:
1) if $|[\mathbf{b}(f_m)]_t| < \tau$ then any update of the tth element increases the cost function; the corresponding element of the solution vector should not be updated;
2) if $|[\mathbf{b}(f_m)]_t| > \tau$ then there exists a non-zero value of the tth element that decreases the cost function and the largest decrement is achieved for the element having the highest value of $|[\mathbf{b}(f_m)]_t|$.
These two statements define the rule for adding a new non-zero element in the solution vector: this is an element having the maximum $|[\mathbf{b}(f_m)]_t|$ subject to $|[\mathbf{b}(f_m)]_t| > \tau$; otherwise, no new non-zero element is added to the solution vector.

We now prove the first statement.

Proof: Let the tth element $[\hat{\mathbf{x}}(f_m)]_t = 0$ be activated as follows: $[\hat{\mathbf{x}'}(f_m)]_t = \alpha = |\alpha|e^{j\arg(\alpha)}$; the updated solution vector is denoted as $\hat{\mathbf{x}}'(f_m)$. The update of the cost function (4.11) is then given by

$$\begin{aligned} \delta &= J[\hat{\mathbf{x}}'(f_m)] - J[\hat{\mathbf{x}}(f_m)] \\ &= \frac{1}{2}|\alpha|^2 - \Re\{\alpha^* \mathbf{a}_t^H(f_m)\mathbf{r}(f_m)\} + \tau|\alpha|. \end{aligned} \tag{4.12}$$

For $|\alpha| > 0$, we have

$$\begin{aligned} \delta &> -\Re\{\alpha^* \mathbf{a}_t^H(f_m)\mathbf{r}(f_m)\} + \tau|\alpha| \\ &> |\alpha|(\tau - |\mathbf{a}_t^H(f_m)\mathbf{r}(f_m)|) \\ &= |\alpha|(\tau - |[\mathbf{b}(f_m)]_t|). \end{aligned} \tag{4.13}$$

Therefore, if $|\alpha| > 0$ then for $|[\mathbf{b}(f_m)]_t| < \tau$, we have $\delta > 0$, i.e. the cost function (4.11) increases. □

We now prove the second statement.

Proof: The cost function is reduced if $\delta < 0$. From (4.12) we can rewrite the update

of the cost function as

$$\delta = |\alpha| \left[\frac{1}{2}|\alpha| - \Re \left\{ e^{-j \arg\{\alpha\}} [\mathbf{b}(f_m)]_t \right\} + \tau \right]. \tag{4.14}$$

For a fixed $|\alpha|$, δ achieves a minimum if $\arg\{\alpha\} = \arg\{[\mathbf{b}(f_m)]_t\}$; in this case,

$$\delta = \frac{1}{2}|\alpha|^2 - |\alpha||[\mathbf{b}(f_m)]_t| + \tau|\alpha|. \tag{4.15}$$

Minimizing δ in (4.15) over $|\alpha|$ and taking into account that now we only consider indexes t for which $|[\mathbf{b}(f_m)]_t| > \tau$, we obtain that the optimal value of $|\alpha|$ is given by

$$|\alpha| = |[\mathbf{b}(f_m)]_t| - \tau$$

which together with $\arg\{\alpha\} = \arg\{[\mathbf{b}(f_m)]_t\}$ results in

$$\delta = -\frac{1}{2} \left[|[\mathbf{b}(f_m)]_t| - \tau \right]^2.$$

Thus, we obtain that there is a value of α that results in $\delta < 0$, i.e. in reducing the cost function. It is seen from the last expression that if we want to add a new element to the solution vector, the index t of the element should correspond to the maximum of $|[\mathbf{b}(f_m)]_t|$ over $t \in \bar{\mathbf{I}}_m(n)$. In this case, we will obtain the largest decrement of the cost function. □

Thus, the value of α that for a fixed t such that $|[\mathbf{b}(f_m)]_t| > \tau$ results in the largest decrement of the cost function is given by

$$\alpha = \left(|[\mathbf{b}(f_m)]_t| - \tau \right) e^{j \arg\{[\mathbf{b}(f_m)]_t\}}.$$

However, this value does not guarantee that the cost function is minimized over the support $\mathbf{I}_m(n+1)$. Therefore, from the above analysis we only use the index t of the element to be added into the support, i.e. we only update the support. The optimization within this support is implemented using the CD search starting from the solution found at the previous iteration.

According to the above statements, the $(n+1)$th non-zero element to be activated in $\hat{\mathbf{x}}(f_m)$ should satisfy

$$k = \arg \max_{t \in \bar{\mathbf{I}}_m(n)} \{ |[\mathbf{b}(f_m)]_t| \} \text{ s.t. } |[\mathbf{b}(f_m)]_k| > \tau. \tag{4.16}$$

After adding the new non-zero element to the solution vector, the CD search, which will be discussed in Section 4.3.3, is used to adjust all elements in $\mathbf{I}_m(n+1)$ to minimize the cost function.

If the problem (4.16) has no solution, the parameter τ is reduced according to the schedule $\tau \leftarrow \lambda\tau$, where λ is a parameter in $(0,1)$. Choosing λ close to one ($\lambda \lesssim 1$) allows a fine sampling of the solutions within $\tau \in [\tau_{\min}, \tau_{\max}]$. The upper limit of τ is set to $\tau_{\max} = \max_t |[\mathbf{c}(f_m)]_t|$, where $\mathbf{c}(f_m) = \mathbf{A}^H(f_m)\mathbf{y}(f_m)$, and $\tau_{\min}$ is a predefined constant. The iterative process stops when the parameter τ is lower than $\tau_{\min}$ or the number of non-zero elements achieves a preset limit S: $||\mathbf{I}_m(n+1)||_0 = S$. The ambiguity surface for localization is then generated from the solution vector $\hat{\mathbf{x}}(f_m)$ as

$$B(\psi_k, f_m) = |[\hat{\mathbf{x}}(f_m)]_k|. \tag{4.17}$$

Throughout this chapter, we use the parameter S to control the process of solving the BPDN problem. Without prior knowledge of the number of true sources, S should be set large enough so that all the sources are located. In the cases that S is larger than the number of true sources, activating an element corresponding to a wrong source position will happen. However, this will not necessarily result in significant performance loss. We can expect that when minimizing the cost function, the signal amplitudes corresponding to the wrong positions will tend to remain small compared to signal amplitudes of the true positions (see numerical results in Section 4.4 that demonstrate this). However, the price paid for choosing S larger than N_s will be a reduced number of true sources that can be detected with N array elements.

4.3.2 Extension to multi-frequency signals

The single frequency ambiguity function can be disturbed by noise and interference. Therefore, the localization obtained by solving the BPDN problem may fail to provide correct source positions. To achieve better localization performance, a frequency diversity can be used. A straightforward method of modifying the above approach to multi-frequency signals is to solve the BPDN problem for each frequency and average the ambiguity surfaces $B(\psi_k, f_m)$, $m = 1, \ldots, M$, over all the frequencies f_m. However, this

combining method may not work properly. This is due to the fact that in the presence of noise, environmental mismatch and interference, different frequencies may give different localizations, resulting in a combined ambiguity surface with many peaks. This may give wrong localization results.

To achieve a greater frequency diversity and avoid the possible strong wrong peaks, we require that, at $(n+1)$th iteration, a non-zero element is added to $\hat{\mathbf{x}}(f_m)$ with the same support for all the frequencies: $I_1(n+1) = \ldots = I_M(n+1) = I(n+1)$. This is done by choosing an element to be added to the support by comparing τ with the peak value of a vector with elements

$$[\mathbf{b}]_t = \frac{1}{M}\sum_{m=1}^{M} |[\mathbf{b}(f_m)]_t|, \;\; t \in \bar{\mathbf{I}}(n). \tag{4.18}$$

The index k of the new element is decided according to the rule:

$$k = \arg\max_{t \in \bar{\mathbf{I}}(n)} \{[\mathbf{b}]_t\} \;\; \text{s.t.} \;\; [\mathbf{b}]_k > \tau. \tag{4.19}$$

The final ambiguity surface is given by

$$B(\psi_k) = \frac{1}{M}\sum_{m=1}^{M} |[\hat{\mathbf{x}}(f_m)]_k|. \tag{4.20}$$

The proposed technique for solving the multi-frequency BPDN problem is summarized in Table 6.3.

As the number of potential source positions K is large, the proposed technique may still result in high complexity. In Section 4.3.4, we describe how to reduce the complexity.

4.3.3 Coordinate descent search

The CD search algorithm is based on the approach in [82] and is summarized in Table 4.2. In order to minimize the cost function for a fixed τ, we apply CD search for each element in $\mathbf{I}(n)$ as shown in Table 6.3. The CD search is performed independently for each of the M frequencies. For an element, four possible updates are considered, corresponding to four Euclidian directions on the complex plane: $\mathbf{e} = [+1, -1, +j, -j]$. For each possible solution update, the cost function update δ is evaluated and if negative, the solution element is updated together with the vector $\mathbf{r}(f_m)$. Specifically, possible updates of element

Table 4.1: Proposed algorithm for solving the complex-valued multi-frequency BPDN problem

Input:

λ, S, $\mathbf{A}(f_m)$, $\mathbf{y}(f_m)$, $m = 1, \ldots, M$

Initialization:

$\hat{\mathbf{x}}(f_m) = \mathbf{0}$, $\mathbf{r}(f_m) = \mathbf{y}(f_m)$

Compute the residual vectors:

$\mathbf{b}(f_m) = \mathbf{A}^H(f_m)\mathbf{r}(f_m), m = 1, \ldots, M$

$[\mathbf{b}]_t = \frac{1}{M}\sum_{m=1}^{M} |[\mathbf{b}(f_m)]_t|, t = 1, \ldots, K$

$k = \arg\max_t \{[\mathbf{b}]_t\}$

Activate the first non-zero element:

$I(1) = k$, $\tau = [\mathbf{b}]_k$, $n = 1$

While $n \leq S$ **and** $\tau > \tau_{\min}$

for $t = I(1), \ldots, I(n)$

CD search for all frequencies

end for

Update the residual vectors:

$\mathbf{b}(f_m) = \mathbf{A}^H(f_m)\mathbf{r}(f_m), m = 1, \ldots, M$

$[\mathbf{b}]_t = \frac{1}{M}\sum_{m=1}^{M} |[\mathbf{b}(f_m)]_t|$

$k = \arg\max_{t \in \bar{\mathbf{I}}(n)} \{[\mathbf{b}]_t\}$

if $[\mathbf{b}]_k > \tau$

Update the support:

$I(i) = I(i-1), i = 2, \ldots, n+1$

$I(1) = k$

$n = n + 1$

else

$\tau = \lambda\tau$

end if

Final ambiguity surface:

$B(\psi_t) = \frac{1}{M}\sum_{m=1}^{M} |[\hat{\mathbf{x}}(f_m)]_t|$ for $t \in \mathbf{I}(n)$ and

$B(\psi_t) = 0$ for $t \in \bar{\mathbf{I}}(n)$

Table 4.2: Coordinate Descent Search

Input:

$t, \tau, \mathbf{a}_t(f_m), \hat{\mathbf{x}}(f_m), \mathbf{r}(f_m), m = 1, \ldots, M$

Initialization:

$h = \tau$

$\mathbf{e} = [+1, -1, +j, -j]$

Repeat Q **times**:

for $m = 1 : M$

$p = 1$

While $(p \leq 4)$

$\alpha = he_p$

$\delta = \frac{1}{2}|\alpha|^2 - \Re\{\alpha^* \mathbf{a}_t^H(f_m)\mathbf{r}(f_m)\}$
$+ \tau(|[\hat{\mathbf{x}}(f_m)]_t + \alpha| - |[\hat{\mathbf{x}}(f_m)]_t|)$

if $\delta < 0$

$[\hat{\mathbf{x}}(f_m)]_t = [\hat{\mathbf{x}}(f_m)]_t + \alpha$

$\mathbf{r}(f_m) = \mathbf{r}(f_m) - \alpha \mathbf{a}_t(f_m)$

else

$p = p + 1$

end if

end While

$h \leftarrow h/2$

end for

Output:

$\hat{\mathbf{x}}(f_m), \mathbf{r}(f_m), m = 1, \ldots, M$

$[\hat{\mathbf{x}}(f_m)]_t$ are given by

$$[\hat{\mathbf{x}}'(f_m)]_t = [\hat{\mathbf{x}}(f_m)]_t + \alpha \tag{4.21}$$

where $\alpha = he_p$, $p = 1, \ldots, 4$. The step size parameter h starts with $h = \tau$ and it is reduced if no update is possible. The reduction schedule $h \leftarrow h/2$ defines the final value of the step size $h = \tau/2^Q$ and, thus, the final accuracy of the solution. In our simulation scenarios, we set $Q = 16$.

4.3.4 Grid Refinement

The proposed technique is based on a predefined grid of potential source positions. For sources distributed in a large area, the number of grid points will be large. This results in high complexity of computing the residual vectors $\mathbf{b}(f_m)$. To reduce the complexity, we explore the idea of grid refinement [79] to reduce the number of potential source positions. This is done as follows.

1) Create grids for the potential source positions at a coarse resolution $(\delta r_i, \delta d_i),\ i = 0$, where δr_i and δd_i are the range and depth step size, respectively.

2) Apply the technique in Table 6.3 to obtain coarse localization results $(r_n, d_n), n = 1, \ldots, S$, and set $i = i + 1$.

3) Build a denser grid around the estimated source positions (r_n, d_n) with a finer resolution $(\delta r_i, \delta d_i)$.

4) Repeat for q times from step 2) until the desired resolution is achieved.

There are different ways of grid refinement. Here, we assume that the fine grid points are equispaced in the local areas around the coarse source positions (r_n, d_n), $n = 1, \ldots, S$. The shape of each local area is defined as a rectangular: $r_k \in [r_n \pm 2\delta r_{i-1}]$ and $d_k \in [d_n \pm 2\delta d_{i-1}]$, where the index k indicates the fine grid points. The resolution at each iteration is enhanced as $\delta r_i \leftarrow \delta r_{i-1}/\gamma_r$ and $\delta d_i \leftarrow \delta d_{i-1}/\gamma_d$, where $\gamma_r > 1$ and $\gamma_d > 1$. In our simulation, a two-step grid refinement process is used by defining the initial resolution at $\delta r_0 = 50$ m and $\delta d_0 = 5$ m and the refinement parameters $\gamma_r = 10$

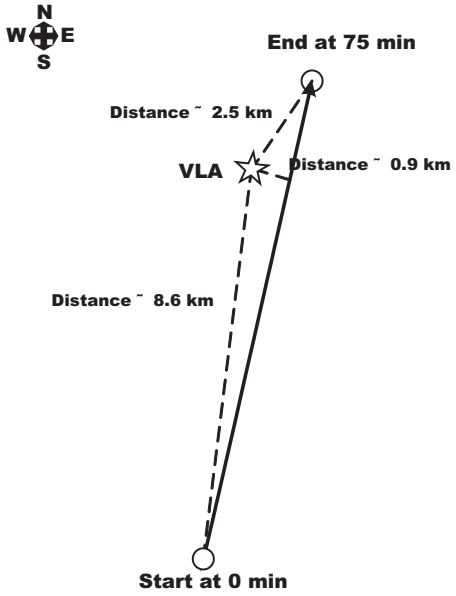


Figure 4.1: Map of the source movement and the location of the receiver hydrophone array.

and $\gamma_d = 5$ to achieve a final resolution of $\delta r_q = 5$ m and $\delta d_q = 1$ m. However, more refinement steps can be used [79].

4.4 Numerical results

In this section, we evaluate the performance of the proposed and known techniques in the environment corresponding to the shallow water underwater acoustic experiment SWellEx-96 (Event S5) [60]. Numerical results obtained from simulated and real experimental data are presented.

The source movement is illustrated in Fig. 4.1. The SSP and the bottom composition parameters from [60] shown in Fig. 3.3 are used to generate the steering vectors $\mathbf{a}_k(f_m),\ k = 1, \ldots, K$. For generation of the steering vectors, the program KRAKEN [86] implementing the normal mode method for computation of the acoustic field is used.

Before applying the proposed method to real experimental data, the performance is evaluated using simulated data.

4.4.1 Simulated data

In the simulation, a set of 13 tones (the same as in the experiment) are transmitted. Below, we use one snapshot for localization, and the duration of the snapshot is 1 s.

In the first simulation scenario, the performance in resolving close sources is evaluated. Signals from two close sources separated by a fixed horizontal distance and both moving at the same speed are generated. The distance from the sources to the receiver antenna is set to 2 km. It has been observed that there is a limit for the horizontal distance between the sources, below which all the MF methods cannot resolve the sources. This can be explained by the waveguide invariant theory [57, 58] stating that

$$\frac{\Delta f}{f} \approx \beta \frac{\Delta r}{r} \tag{4.22}$$

where Δf is a shift of frequency f in the channel frequency response magnitude due to horizontal movement of the transmitter from distance r to $r + \Delta r$, and $\beta \approx 1$ is the waveguide invariant. In this simulation scenario, the delay spread of the channel impulse response is approximately 0.2 s, resulting in a coherence bandwidth of the channel $f_{\text{coh}} = 5$ Hz. To distinguish two different sources at a distance Δr, the frequency shift Δf should be higher than the coherence bandwidth: $\Delta f \gg f_{\text{coh}}$. From (4.22), we obtain $\Delta r \gg 25$ m for a tone of 388 Hz and $\Delta r \gg 200$ m for a tone of 49 Hz. By increasing the distance Δr between the two sources, more and more frequencies start to resolve the sources. In the simulation, it has been found that for a fixed horizontal distance of $\Delta r = 80$ m, the BPDN method with and without grid refinement can provide reliable results in resolving the sources (with 100% successful rate). In the same scenario, the conventional MF and IAA techniques resolve the two sources at as low rates as 19% and 24%, respectively.

The second simulation scenario is aimed to test the ability of the proposed method in detecting multiple sources. To avoid the error caused by sources with very small distance, the sources are assumed to have horizontal distance of at least 75 m or a depth difference of at least 5 m from each other. The source positions are drawn randomly with a uniform distribution from a range interval of $[7.3, 8.7]$ km and depth interval of $[20, 180]$ m. Fig.4.2 shows the localization results in terms of probability of localization, which is calculated as the ratio between successfully detected positions (with range and depth errors smaller than some thresholds) and the total number of sources N_s. The detected positions are

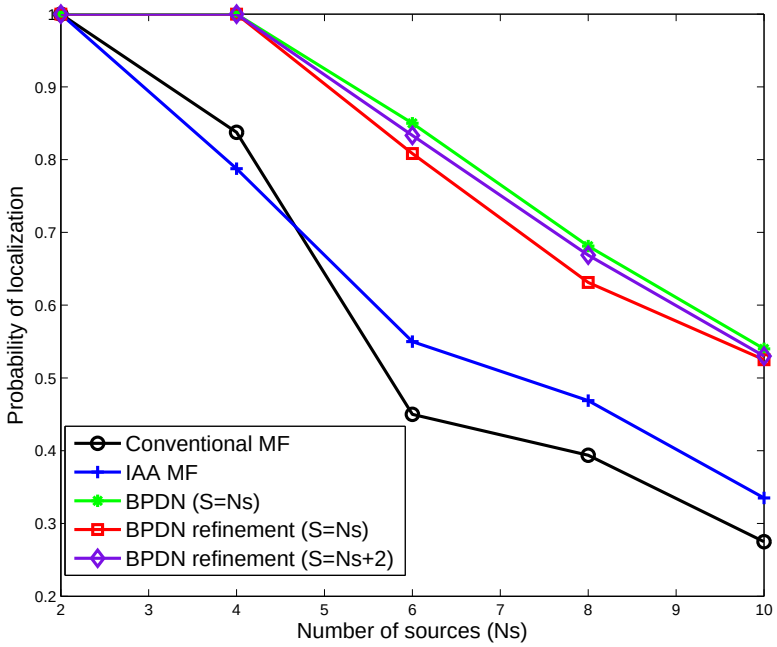


Figure 4.2: Probability of successful localization *vs* the number of sources.

positions of N_s greatest maximums of the ambiguity surface. A position is considered successfully detected if its range and depth differ from the true range and depth less than 25 m and 2 m, respectively. The results shown in Fig.4.2 are obtained by averaging over 20 simulation trials.

It is seen that the IAA MF processor has a slightly better performance than the conventional MF processor. However, both these methods have significantly worse performance than the proposed method (the proposed method can locate more sources simultaneously). Introducing the grid refinement for solving the BPDN problem results in an inferior performance compared to the BPDN solution on the originally fine grid, if the parameter S is set to equal to the true number of sources $S = N_s$. This is due to the fact that with the coarse grid, the steering vector mismatch is large. When solving the BPDN problem using the coarse grid, some true positions may be missed out, whereas, some wrong positions are picked up into the support. Increasing S at the coarse grid allows resolving this problem. As seen in Fig.4.2, setting $S = N_s + 2$ at the coarse grid results in almost the same performance as that of the BPDN on the fine grid.

The third simulation scenario tests the robustness of the proposed method to strong

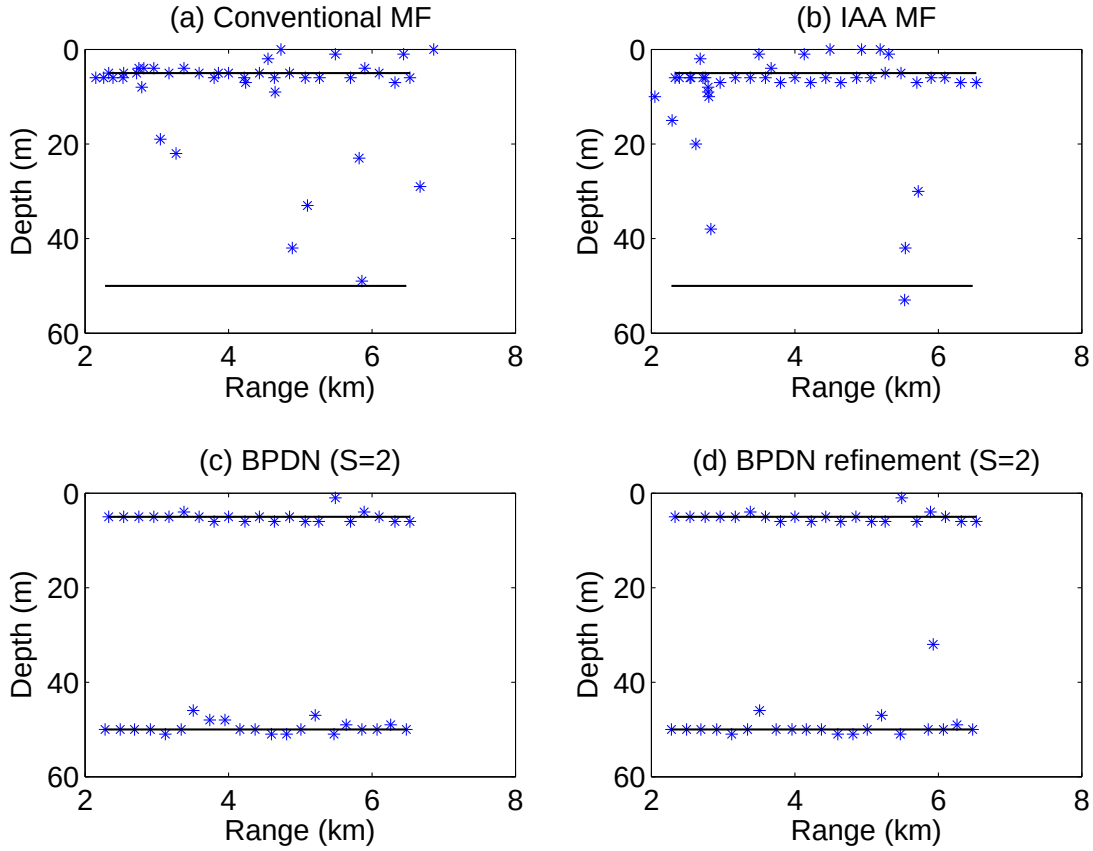


Figure 4.3: Localization results for the source of interest at a depth of 50 m with a strong interfering source at a depth of 4 m. Signal to interference ratio is -10 dB. The solid lines are the trajectories of the two sources and '*' denotes the estimated positions.

interference from the sea surface and noise. This is a scenario where a source of interest is towed at a depth of 50 m by a moving ship. The interference is modeled as a source at a depth of 4 m transmitting the same 13 tones at a power 10 times as that of the source of interest. White noise is added to the received signal, resulting in an SNR of 10 dB. Fig.4.3 shows localization results obtained by the four methods. The true moving trajectory is indicated as a solid line. It is seen that the proposed method provides reliable detection of the source of interest, whereas the conventional MF and IAA MF techniques fail to locate the weak source of interest due to the strong interference. For both the BPDNs, we have here $S = 2$.

The fourth simulation scenario combines the effects of noise, interference, multiple and close sources. In the simulation, four acoustic sources are involved. All the sources

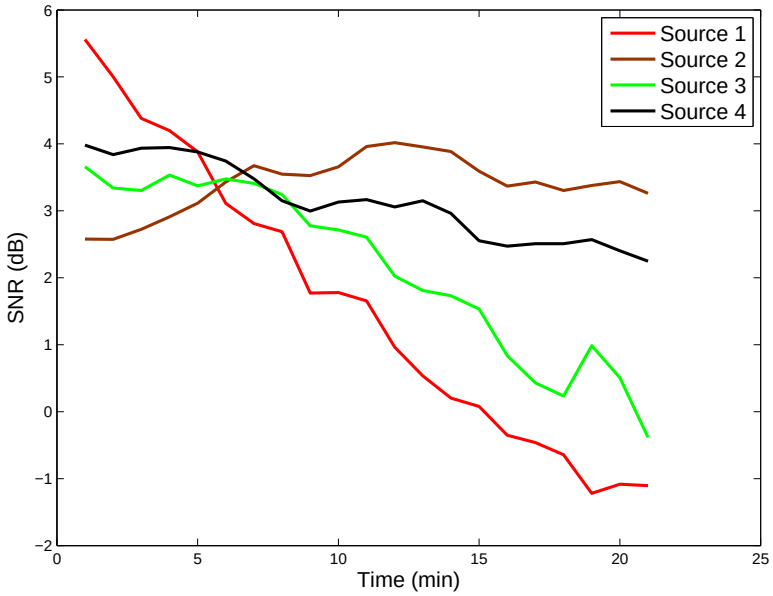


Figure 4.4: SNR for each of the four sources corresponding to the results shown in Fig.4.5 and Fig.4.6: the sources are moving in the range interval $[7.3, 8.7]$ km and the depth interval $[20, 180]$ m.

are transmitting signals simultaneously at the same power. The sources are moving in a range interval within $[2, 7]$ km and depth interval $[0, 200]$ m with different speeds and directions. The first source moves horizontally away from the receiver at a speed of 3.5 m/s. The second source moves vertically towards the sea bottom at a speed of 0.125 m/s. The third source moves with a horizontal speed of 1.5 m/s away from the receiver and also has a vertical speed of 0.125 m/s towards the sea surface. The forth source moves away from the receiver and towards the surface with horizontal and vertical speeds of 1.0 m/s and 0.025 m/s, respectively. White noise is added to the simulated signal resulting in SNRs as shown in Fig.4.4, where the SNR of each source is calculated as the ratio between the received signal power of the specific source and the noise power.

Fig.4.5 and Fig.4.6 show localization results obtained by the four methods. The colored solid lines are the trajectories of the four sources; points marked by '*' are the localization results. The measurements are obtained every 1 min using a 1-s snapshot. It is seen in Fig.4.5(a) that the localization results obtained by the MF processor are partially scattered. For most of the measurements, the conventional MF processor can provide accurate source locations for two or three of the sources. However, it fails to provide

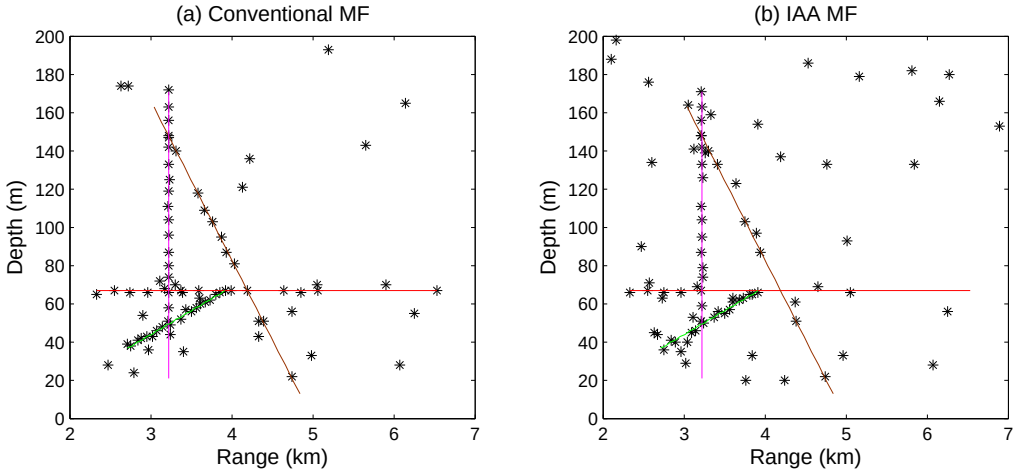


Figure 4.5: Localization results in simulation for four moving sources: the solid lines are the trajectories and '*' denotes the estimated source positions. The SNR for each individual source is shown in Fig.4.4.

accurate localization for all four sources simultaneously. The localization performance of the IAA MF processor, in this scenario, is worse than that of the MF processor. This is due to a small number of snapshots (one snapshot) resulting in inaccurate estimation of the matrices $\mathbf{R}(f_m)$. The probability of localization for the conventional MF and IAA MF processors are 71% and 56%, respectively.

Compared with the conventional MF and IAA MF processors, the proposed method performs significantly better. It is seen in Fig.4.6(a) and Fig.4.6(b) that the localized positions are confined to the trajectories well, providing probability of localization of 96% and 91%, respectively. A performance loss is observed due to introducing the grid refinement when solving the BPDN problem. By increasing S beyond the number of true sources ($S = 6$), the performance is improved; the probability of localization is now 96% (see Fig.4.6(c)), i.e. the same as in the original BPDN.

So far, we considered the localization with perfectly known environmental parameters. In practice, a mismatch between true and known environment always exists that can downgrade the performance of MF localization [87]. Next, we investigate the sensitivity of the proposed technique to the mismatch using an example. Fig. 4.7 shows the true and mismatched sound speed profiles. The source positions are the same as shown in Fig. 4.6. The BPDN technique without the grid refinement provides a probability of localization at

85%. For the BPDN with grid refinement, the probabilities of localization are 81% and 85% for $S = 4$ and $S = 6$, respectively. A performance loss is observed compared with the case of perfectly known environment. However, the probability of localization is still higher than that provided by the conventional MF and IAA MF processors with perfectly known environment.

4.4.2 Experimental data

We now apply the four localization methods to real experimental data. To imitate the multi-source signals (i.e. a snapshot generated by several sources), we combine snapshots from the same source recorded at different positions. The snapshot used to generate results in Fig.4.8 is a combination of two snapshots, one of which is taken at a distance of 3.11 km (see Fig.4.1) and the other is taken from a distance of 3.18 km. Note that the noise level of the resulting multi-source signal is higher than that in the original snapshot, i.e. the SNR is reduced with increasing the number of sources.

Due to the source movement, the received signals suffer from a Doppler shift. Therefore, frequency estimation is required to capture the information on the shifted frequencies to provide accurate extraction of the signals. We adopt the frequency estimation method [56] as explained in [61] to estimate the received tone frequencies and the signals are extracted using the discrete Fourier transform. The snapshot length is set to 2 sec.

Fig.4.8 shows ambiguity surfaces of the four methods for two close sources. The two sources are about $60 - 75$ m away from each other. The ambiguity surfaces are normalized with respect to the maximum values; values below a threshold of 0.1 of the maximum are all set to value of the threshold so that the dynamic range of the ambiguity surface is from -10 dB to 0 dB. In Fig.4.8(a)(b), the conventional MF and IAA MF processors fail to separate the sources, although the IAA MF processor provides lower sidelobes. The ambiguity surfaces generated by the proposed method provide clear and accurate indication of the source positions. The results provided by the BPDN with the grid refinement are exactly the same as that of the original BPDN.

Fig.4.9 shows the ambiguity surfaces obtained for a scenario with three sources. In this

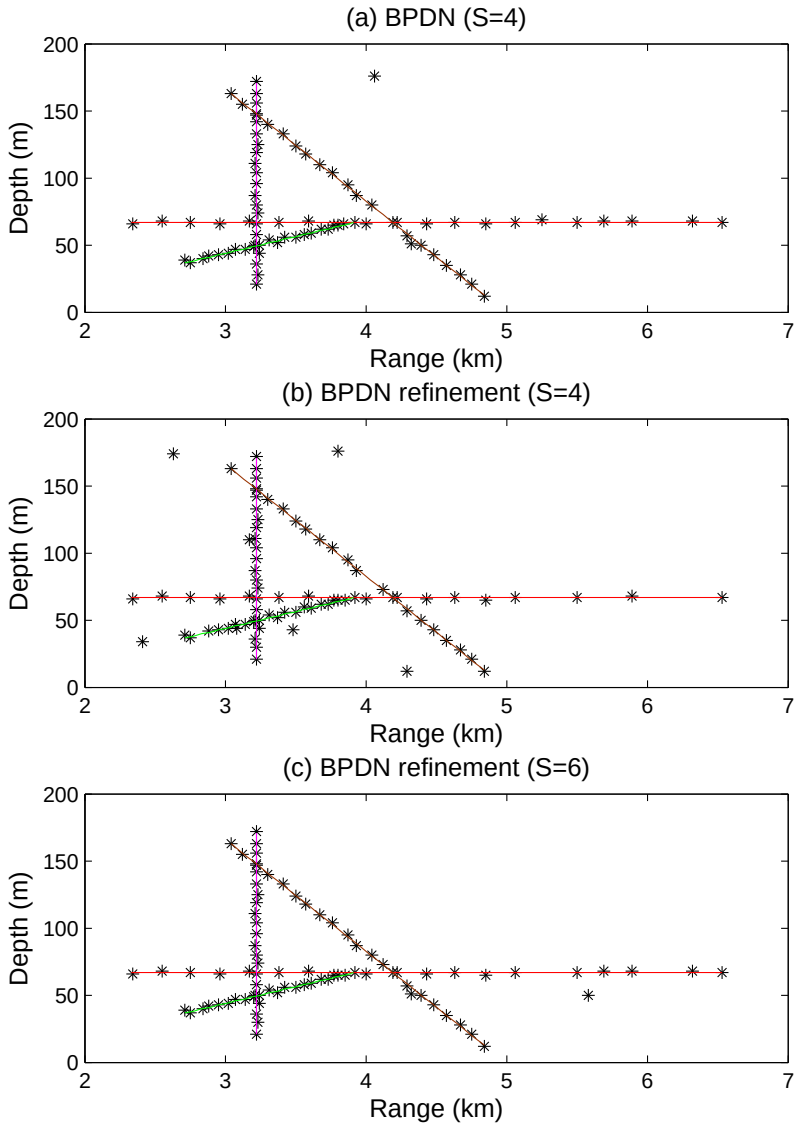


Figure 4.6: Localization results in simulation for four moving sources: the solid lines are the trajectories and '*' denotes the estimated source positions. The SNR for each individual source is shown in Fig.4.4.

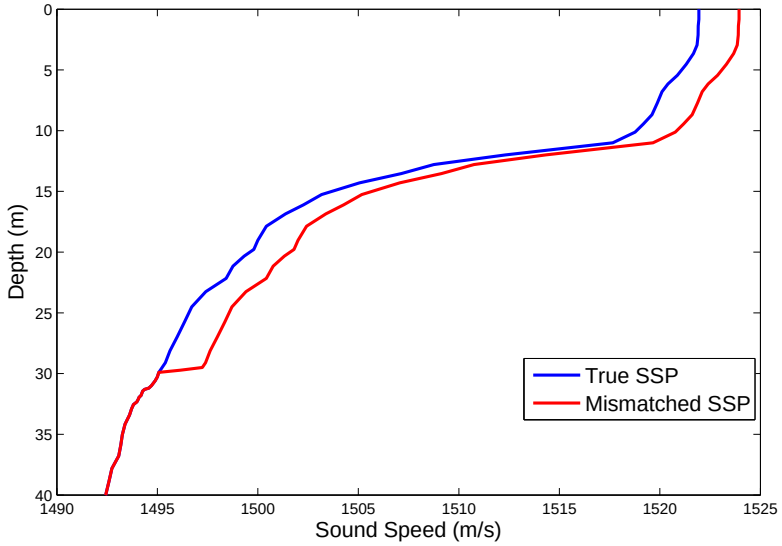


Figure 4.7: Mismatched ans true sound speed profile.

case, the noise level is further increased due to combining three noisy snapshots. Both the conventional MF processor and the IAA MF processor fail to locate the sources; there are many high peaks which inaccurately indicate the source positions. However, as seen in Fig.4.9(c)(d), the three source positions found by the proposed method are shown clearly and accurately. The result obtained with the grid refinement is very close to that obtained by the BPDN on the fine grid.

4.5 Summary

In this chapter, we have formulated the underwater acoustic source localization in the framework of the MF processing as a sparse representation problem and demonstrated that its solution based on the complex-valued multi-frequency BPDN shows high localization performance. For solving the BPDN problem, we have used the Homotopy approach combined with the coordinate descent search. Grid refinement has further been used to reduce the complexity of solving the BPDN problem. Comprehensive simulation has been carried out to evaluate the performance of the proposed method in resolving sources close to each other, locating multiple sources and robustness to noise and interference. The effectiveness of the proposed method has also been verified using real ocean

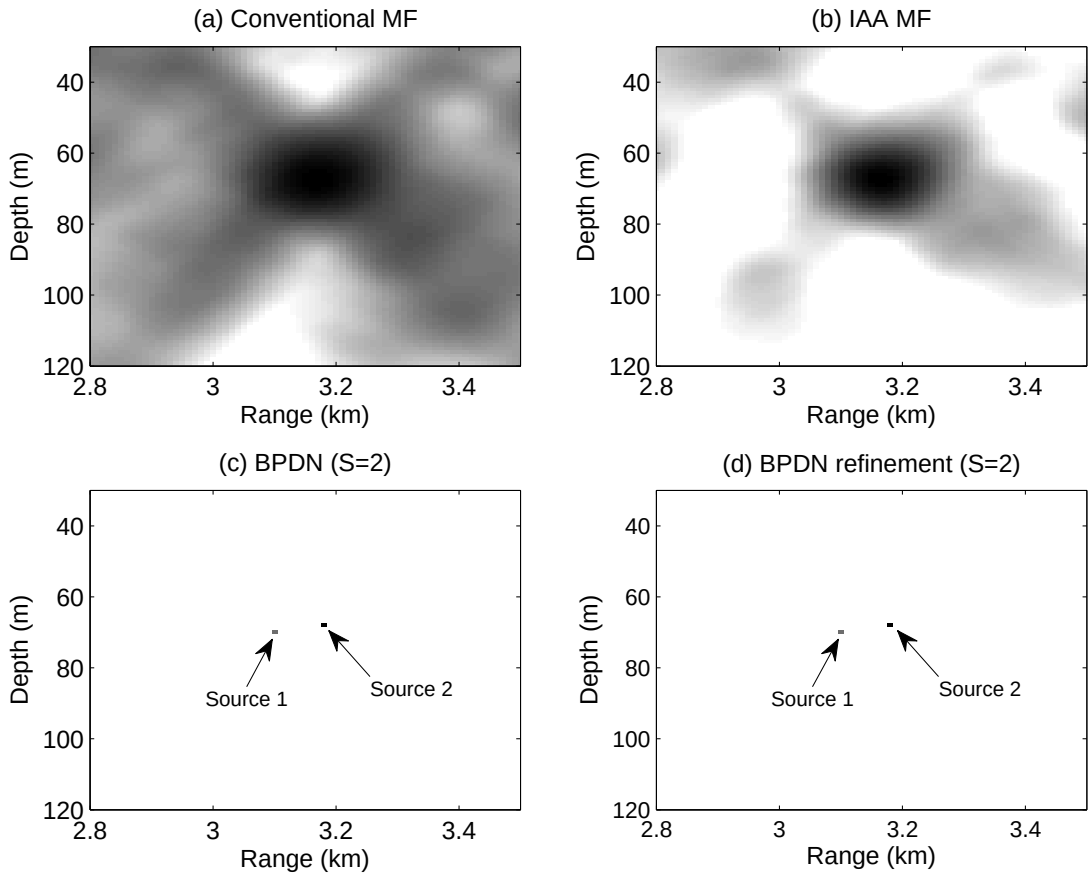


Figure 4.8: Ambiguity surfaces for localizing two close sources at [3.11 km, 70 m] and [3.18 km, 68 m] (Source locations for BPDN methods are indicated by arrows).

experimental data. It has been shown that the proposed technique outperforms other MF techniques, such as the conventional MF processor and IAA MF processor, in resolving sources positioned closely to each other, tolerance to noise and capability of locating multiple sources.

MF source localization techniques require accurate knowledge of underwater environment. In the next chapter, we will introduce an MF inversion technique for estimating unknown environmental parameters. As mentioned at the beginning of this chapter, knowledge of source positions is useful for improving system performance. In Chapter 6, we will investigate the use of channel sparsity, which can be estimated from the transmitters and receivers positions, to improve the channel estimation accuracy and reduce the complexity.

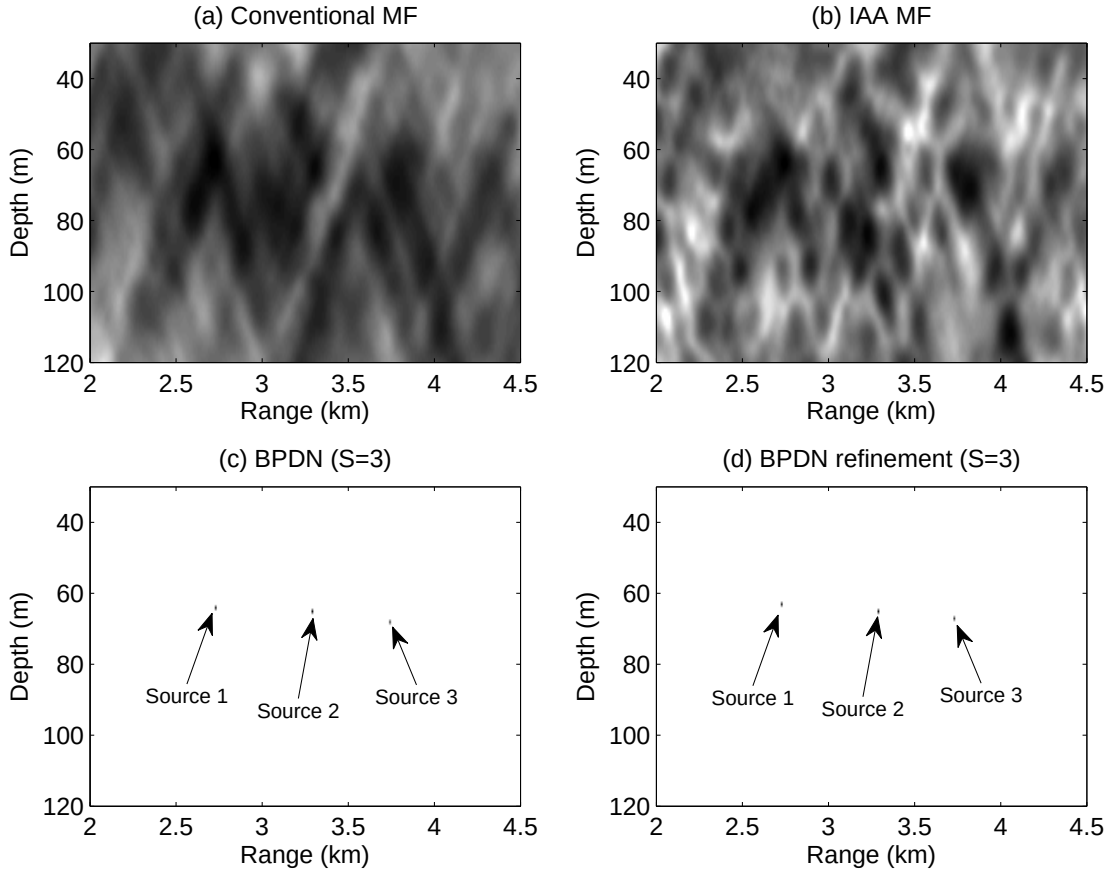


Figure 4.9: Ambiguity surfaces for localizing three sources at [2.73 km, 63 m], [3.3 km, 65 m] and [3.74 km, 68 m] (Source locations for BPDN methods are indicated by arrows).

Chapter 5

Estimation of Sound Speed Profile using Matched Field Inversion

Contents

As described in Chapter 4, underwater sources can be localized using MF techniques provided that the underwater acoustic environment is known. However, due to environment mismatch caused by variations of sound speed profile (SSP) in the water column and inaccurate knowledge of geoacoustics parameters, the accuracy of the localization degrades. MFI can be used for estimating the environmental parameters.

5.1 Introduction

In underwater acoustic signal processing, matched field processing is an important approach which is widely used for source localization [14] and MFI problems [88–95].

In this approach, acoustic field replicas generated by a numerical underwater acoustic propagation model with candidates of unknown parameters are matched to the measured acoustic field at the receiver hydrophones by a proper cost function. The two problems are similar in that by searching within a parameter range, the parameters with the highest match (lowest mismatch) are selected as estimates of the unknown parameters. For the localization problem, the parameters are the range and depth of an acoustic source. For the MFI problem, the parameters are bottom parameters and uncertain sound speed values at different depths. The localization problem can usually be solved by exhaustive search. However, for MFI the exhaustive search is infeasible and, due to the nonconvex cost function, a global optimization method must be used for searching the unknown parameters. SA [88, 91, 92, 94] and genetic algorithms [93, 95] are the most popular global search methods for such problems. Extensive work on the inversion of bottom parameters in shallow water have been done and positive results have been achieved [88, 91, 92, 94, 95]. In deep water scenarios, representation of the SSP tends to need more parameters than that in shallow water. Thus, more computation is required for the inversion of whole SSP in deep water. However, the variations of the sound speed mostly concentrated at shallow depth, therefore the inversion is applicable by spending less computation for deeper depth.

In this chapter, we apply a global search method to solve an MFI problem for estimating the SSP in a deep water scenario. We assume that the SSP is a range independent Munk profile [27]. An adaptive hybrid global optimization method [88] that combines the downhill simplex (local optimization) method and simulated annealing (global optimization) method is used to search the parameters.

5.2 Simulated annealing

SA is a global optimization method arisen from the condensed matter physics [96]. In condensed matter physics, the behavior of the matter at low limit temperature is a fundamental question. The behavior is mostly determined by the lowest energy state (ground state), as described in [96]. Therefore, finding the ground state of the atom is essential, which is termed as annealing. An example in [96] is very useful for understanding the

idea behind the SA algorithm: in physics experiments, to grow a crystal from a melt, the solid should be melt (heated) first and then slowly cooling down. Without this, the resulting matter may be with no crystalline order, which are locally optimal structures.

The SA algorithm is similar to the process of this experiment, which also has a gradually and carefully reduced temperature to control the searching process. It is developed to make the iterations escape from local optimum and finally reach the global optimum. To apply the SA algorithm to an MFI problem, a cost function should be defined at the beginning. As mentioned, MFI is based on MF processing which uses global search algorithms to explore the unknown parameter space. Estimation of the unknown parameters is obtained when the mismatch between measured field and the field replica is generated according to tentative parameters. The cost function used to calculate the mismatch should be sensitive to the mismatch of the parameters and easily to compute. Due to the heavy computation caused by the random search, we choose the Bartlett linear processor [14] to construct the cost function:

$$E = 1 - \mathbf{w}^H \mathbf{R} \mathbf{w} \tag{5.1}$$

where $\mathbf{w}$ is the field replica vector at the receiver array and $\mathbf{R}$ is the cross-spectral matrix of the measured field:

$$\mathbf{R} = \mathrm{E}\{\mathbf{v}\mathbf{v}^H\} \tag{5.2}$$

where $\mathbf{v}$ is the measured field vector at the receiver array. All the field vectors are normalized so that they have a unit length.

Based on the cost function (5.1), the global optimum is the global minimum. The core idea of SA is: Perturb the initial parameters, which are random values within the search range, and compute the difference $\triangle E$ between the new value of the cost function and the initial one. The perturbation of the ith parameter m_i, in general, can be expressed by:

$$m_i = m_{i'} + \zeta_i \tag{5.3}$$

where $m_{i'}$ is the ith parameter of previous state and ζ_i is a random parameter making the perturbed parameter m_i fall in the interval $[m_i^-, m_i^+]$. Parameters m_i^- and m_i^+ are the lower and upper bound of the ith parameter.

If $\triangle E \leq 0$, the new state is accepted unconditionally. For the case $\triangle E > 0$, the

probability of accepting the state is given by [96]:

$$P(\triangle E) = e^{-\frac{\triangle E}{k_B T}} \tag{5.4}$$

where k_B is the Boltzmann's constant and T is the critical temperature.

In the case that the new state has smaller mismatch, the process is moving downhill (towards either the global or one local minimum). While in the other case, the process has some probability to move uphill so that the state can escape from local minimum. At the beginning of the searching, the critical temperature is set to a large value and the search is mostly random. As T is reduced, the probability of accepting a perturbation with an increased mismatch $\triangle E > 0$ is reduced. However, at nonzero temperature, it is always possible to escape from local minimum. For conventional SA, to guarantee convergence to the global minimum,the cooling schedule should satisfy [97]:

$$T_n = T_0/log(n) \tag{5.5}$$

where T_0 is the initial temperature and n is the time order (the order of steps reducing the temperature).

The SA algorithm is an effective method to solve global optimization problems, however, it still needs quite significant computation. To reduce the cost of computation and make a faster annealing schedule, significant amount of research has been done. Two methods can be used to control the SA: one is to control the perturbing algorithms and the other one is to control the cooling schedule of the critical temperature.

In [98], Harold Szu and Ralph Hartley proposed the FSA algorithm, where the Cauchy distribution is used to control the random parameter search. The perturbation of the ith parameters m_i is given by [91]:

$$\zeta_i = \xi \triangle_i \tag{5.6}$$

where $\triangle_i$ is a random variable uniformly distributed in the same range as ζ_i in (5.3) and ξ is a Cauchy-distributed random variable defined by:

$$\xi = \left(\frac{T_n}{T_0}\right)^{1/2} tan\left[\pi\left(\eta - \frac{1}{2}\right)\right] \tag{5.7}$$

where η is uniformly distributed in [0,1]. For the FSA, the perturbed parameters can be expressed by:

$$m_i = m_{i'} + \left(\frac{T_n}{T_0}\right)^{1/2} tan\left[\pi\left(\eta - \frac{1}{2}\right)\right] \triangle_i \tag{5.8}$$

According to (5.8), the perturbed parameter m_i follows a Cauchy-distribution centered at parameters of the previous state $m_{i'}$. With a high temperature, the probability of a large perturbation (large ξ) is high, which is important to escape from local minimum. For lower temperatures, the perturbations tend to be small, e.g., the random search is mostly carried out around the current value. However, a large perturbation is still possible.

In FSA, the cooling of the temperature is given by [98]:

$$T_n = \beta^n T_0 \tag{5.9}$$

where β is a positive constant less than one.

Another SA algorithm of interest was proposed by Lindsay and Chapman and examined in the bottom inversion problem of shallow water in [92]. They used an adaptive algorithm to cool down the critical temperature. The perturbation is different from that of the FSA in that it only perturbs one of the parameters which has the highest sensitivity to the perturbation according to the previous iterations. In this adaptive SA, some iterations should be done initially to provide the estimation of the sensitivity of each parameter. The accepting probability is fixed and the critical temperature is dependent on the sensitivity. As the sensitivity decreases, the temperature decreases too. To enable convergence, the search ranges of the parameters are reduced slowly according to the previous perturbations. Also, the random search is based on the previous iterations: parameters with smaller mismatch tend to have a larger probability to be searched. The convergence is accelerated by adopting the experience in the iterations.

5.3 Hybrid simulated annealing algorithms

The main problem of SA algorithms is the high computation cost. This high cost is due to the random search of the parameters with little gradient information. Compared to local optimization methods, the global optimization tends to converge much slower. Therefore, faster algorithms that combine both the local optimization and global optimization methods are proposed. In this section, two hybrid algorithms combining the DHS method and SA are introduced against the inversion problem, which will be used in our simulation. To start with, we will firstly introduce the DHS.

5.3.1 Downhill simplex

The DHS method [99] is a local optimization method for minimizing a multiple parameter function. It is useful in the hybrid algorithms in that it is simple and efficient and there is no need to compute the gradient [91].

In a scenario where the M unknown parameters to be estimated are represented by a vector (model) $\mathbf{m}$, $M+1$ models are needed to construct a simplex for DHS. Each of the models $\mathbf{m}_k$ is weighted by a mismatch $E_k, k = 1, \ldots, M+1$. The algorithm can be divided into the following steps [99]:

1. Randomly choose $M+1$ models and calculate the corresponding mismatch E_k, $k = 1, \ldots, M+1$. The model with the highest (lowest mismatch) is $\mathbf{m}_h$ ($\mathbf{m}_l$), the corresponding mismatch is denoted as E_h (E_l).

2. Compute the centroid of the models except the model $\mathbf{m}_h$, which is given by:

$$\mathbf{m}_c = \frac{\sum_{k \neq h} \mathbf{m}_k}{M}, k = 1, \ldots, M+1. \tag{5.10}$$

3. Compute the reflection of $\mathbf{m}_h$ from the centroid $\mathbf{m}_c$, and calculate the corresponding mismatch E_r. The reflection can be expressed, as described in [91], by:

$$\mathbf{m_r} = \mathbf{m}_c - (\mathbf{m}_h - \mathbf{m}_c). \tag{5.11}$$

4. Compare the reflection mismatch E_r with the mismatches of the initial simplex:

- If $E_r < E_h$ and $E_r > E_l$, the model $\mathbf{m}_h$ with the highest mismatch is replaced by the reflection model $\mathbf{m}_r$.

- If $E_r < E_l$, an extension, $\mathbf{m_{rr}}$ to the reflection model is performed, which is given by:

$$\mathbf{m}_{rr} = \mathbf{m}_c - 2\left(\mathbf{m}_h - \mathbf{m}_c\right). \tag{5.12}$$

If the expansion mismatch E_{rr} is still smaller than the lowest mismatch, then the highest model is replaced by the expansion model. Otherwise, the expansion is failed and the highest model is replaced by the reflection model.

- If $E_r > E_h$, a contraction of $\mathbf{m}_h$ towards $\mathbf{m}_l$ is carried out and will be accepted if lower than the highest mismatch. The contraction is given by:

$$\mathbf{m}'_h = \mathbf{m}_h - \frac{\mathbf{m}_h - \mathbf{m}_l}{2}. \tag{5.13}$$

Otherwise, a multiple contraction is made on all the models of the current simplex models towards the model $\mathbf{m}_l$, given by the following equation:

$$\mathbf{m}'_k = \mathbf{m}_k - \frac{\mathbf{m}_k - \mathbf{m}_l}{2}, k \neq l. \tag{5.14}$$

At each DHS step, one new model is generated for reflection, extension to reflection and single contraction while M new models are generated for multiple contraction. After these steps, the new models will be used as the initial of the next iteration. Any resulting parameter values outside the search range will be replaced by the nearest bound. As the steps are repeated, the difference between the highest mismatch and lowest mismatch will become smaller and smaller and finally converge to a local minima. Usually, the DHS will stop when the difference between the lowest and the highest mismatch is reasonably small, depending on different problems. The criteria can be expressed in the following function [91]:

$$\frac{E_h - E_l}{(E_h + E_l)/2} < \epsilon, \tag{5.15}$$

where ϵ is the tolerance.

5.3.2 Simplex simulated annealing (SSA)

The simplex simulated annealing(SSA) [91] is a hybrid version of the DHS and FSA, which incorporates perturbation into DHS steps by performing all DHS steps on perturbed models. FSA provides possibility of escaping from local minimum to DHS, while

DHS provides additional gradient information to accelerate the searching process to FSA. Another benefit of the hybrid algorithm is that best models with lowest mismatch are always retained since, at each iteration, only the models with higher mismatches will be updated. Figure 5.1 is a block diagram of the SSA.

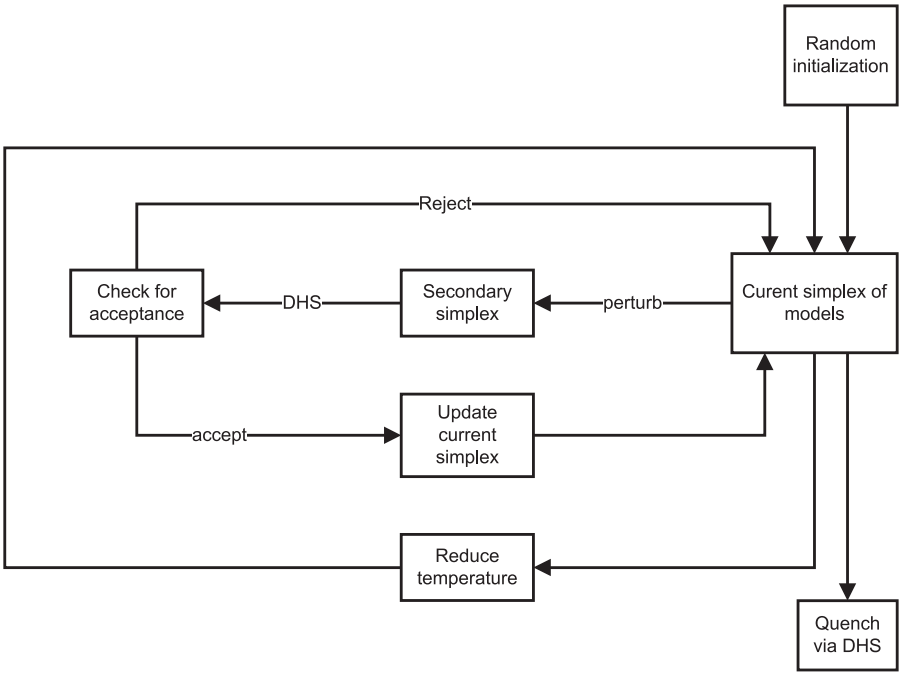


Figure 5.1: Block diagram of SSA

DHS steps are carried out after random perturbations to the models of simplex. The perturbation of the models is different from that of the FSA in that both the parameters to be estimated and the mismatches are perturbed. The procedure of SSA is divided into the following steps:

1. Initialize the models of simplex by random selection within the search range to form the current simplex.

2. Perturb each of the models of the simplex, including the parameters and the mismatches, to construct the secondary simplex of models. The perturbation of the parameters is given by (5.8) and the perturbation of the mismatch is similar, which is given by:

$$E = E' + \xi\overline{E} \tag{5.16}$$

where E' is the current mismatch and $\overline{E}$ is the average mismatch of current models of simplex. E is the resulted mismatch of the perturbed simplex (secondary simplex).

3. Perform the DHS steps on the secondary simplex and use (5.4) to check whether to accept the new models. The difference of the mismatch is calculated between that of the new model generated by each DHS step and the model in the current simplex. Usually within one iteration, there are several DHS steps to be performed before reducing the temperature and starting the next iteration.

Processing at high temperature is dominated by random search since large perturbation happens and the steps of uphill are accepted with high probability. At low temperature, the acceptance of uphill steps is low, therefore the processing is more of DHS. This algorithm will highly increase the efficiency of the processing in that at low temperature, since considerable amount of iterations have been done, it is more reliable that the global optimization is within or close to the range of the models of simplex. Therefore, the problem is more likely to behave as a local optimization problem.

4. Quench [88] the iterations by stopping the random perturbation and using DHS steps only. When the designed number of temperature steps have been done, the random search is stopped and the model with lowest mismatch is selected out and saved. All other models are replaced by random models within the search range. The criteria in (5.15) is used to control the quenching. This quench method is a double check for the best result and is possible to produce better results.

5.3.3 Adaptive simplex simulated annealing (ASSA)

The adaptive simplex simulated annealing (ASSA) algorithm is also a hybrid version of DHS and FSA, which incorporates the random perturbation in FSA into each DHS step. The algorithm firstly performs one DHS step on the current simplex of models. Each of the new models generated by DHS steps is perturbed and checked to update the current simplex of models. The scaling of the perturbation distribution is based on the

average size of the previous accepted steps, which will provide a rough estimation of the perturbation range with a reasonable acceptance rate.

The procedure of ASSA is somewhat similar to that of SSA, which is described in what follows [91]:

1. Initialize the models by using the same method described in SSA.

2. DHS steps are performed on the current simplex models, which are the initial models at the very beginning of the process and are updated after perturbations. At this stage, new models are generated which will be perturbed and checked later. The current simplex of models is not updated.

3. Random components are introduced to DHS steps by perturbing the new generated models. For ASSA, all parameters of the new models are perturbed at one perturbation step and the mismatch of perturbed model is calculated and compared with the current models for acceptance. The perturbation of each parameter is based on Cauchy distribution centered at current parameter values, which is given by:

$$m'_k(i) = m_k(i) + \zeta \triangle_k(i), i = 1, \ldots, M, \tag{5.17}$$

where $\triangle_k(i)$ is the distribution width of the ith parameter and ζ is a random variable that follows Cauchy distribution, given by:

$$\zeta = \eta_1 \tan\left[\pi\eta_2/2\right], \tag{5.18}$$

where η_1 and η_2 are uniform random variables on $[-1, 1]$.

The distribution described in (5.18) generates variables mostly within $[-1, 1]$, thus the random perturbations are mostly smaller than the distribution width. Therefore, the initial distribution width should be large enough so that the perturbations are not limited and the parameter space is well sampled and characterized by accepted perturbations. The initial distribution width of each unknown parameter is set to the difference between the upper limit and the lower limit of this parameter initially. Then it is adaptively scaled according

to the average size of recently accepted perturbations, which provides indication of the ranges of random perturbations with a reasonable acceptance rate. To more precisely define the adaptively scaling algorithm, the distribution width is set to a factor s times the average absolute value of the recent S accepted perturbations. S is set to 30 in our tests and s to 2 to provide a reasonably fast speed of convergence but still quite accurate results.

After the perturbations on the new models, the resulting models are checked for acceptance. The criteria for acceptance is the same as that of SSA. The accepted models will always replace the model with highest mismatch in the current simplex and then another iteration continues. There are usually several iterations at one temperature step to provide good performance on guiding the search to the global minimum and escaping from a local minimum.

4. When the desired number of temperature steps are finished, a quenching, as described for SSA, is carried out.

For both SSA and ASSA algorithm, the initial critical temperature T_0 should be set carefully so that most perturbations are accepted at the beginning. The parameter β should also be chosen carefully so that the annealing schedule provides a balance between fast convergence and accurate global optimum. Through our simulations, the initial critical temperature is set to $T_0 = 0.3$ and the reduction factor is varied in different cases.

5.4 Simulation results

In the previous section, we have introduced two hybrid versions of simulated annealing algorithms. In this section, we firstly compare these two methods by optimizing a multimodal function. Then the results of applications on SSP inversion are shown.

5.4.1 Comparison of ASSA and SSA

ASSA perturb parameters based on previous accepted perturbation which will give an indication of the range of random search within which a perturbation is accepted with a reasonable probability. In this sense, ASSA will be more efficient than SSA. In this part, we will examine the two methods by using a known multimodal function, which is given by [91]:

$$\begin{aligned} E(x_1, x_2, x_3, x_4, x_5, x_6) = \; & 4.8 + x_1^2 + 5x_2^2 + 0.1x_3^2 + 0.05x_4^2 + x_5^2 + x_6^2 \\ & -0.03\cos 4\pi(x_1 - x_2) + 1.4\cos 4\pi(x_1 + x_2) \\ & -0.5\cos 10\pi(0.05x_4 - 0.1x_3) \\ & -1.0\cos 10\pi(0.05x_4 + 0.1x_3) - 0.25\cos 5\pi(x_5 - x_6) \\ & -1.35\cos 5\pi(x_5 + x_6) \end{aligned} \tag{5.19}$$

The function has a global minimum 0 when every variable is 0. We adopt two criterions to define a successful inversion for the function, which are given by:

$$x_i \leq \epsilon_1, i = 1, \ldots, 6, \tag{5.20}$$

$$E \leq \epsilon_2, \tag{5.21}$$

where ϵ_1 and ϵ_1 are small values close to zero. In the simulation, the parameters used to control the searching and cooling algorithm are set to the same values for the two methods: β is set to 0.995, and 2000 temperature steps are used before quenching. The maximum number of random steps at each temperature is set to 60 and the number of accepted iterations at each temperature is set to 5. For each method, 10 independent simulations are performed. Results of mismatches and the total number of objective function evaluations are shown in Table 5.1. We find that with this relatively slow cooling schedule, ASSA can find the global minimum much more accurately than SSA despite that the convergence to the global minimum is not guaranteed (with one unsuccessful inversion out of ten in total). However, for most of the test cases, ASSA has much better performance than SSA, with a function value at the level of $\epsilon_1 = 10^{-7}$ compared to $\epsilon_1 = 10^{-4}$ and variable values at the level of $\epsilon_2 = 10^{-3}$ to $\epsilon_2 = 10^{-2}$.

Furthermore, even with much better results than SSA, ASSA tends to need a much smaller number of the objective function evaluation, which is 5.08×10^4 averaged in our

Table 5.1: Comparison between ASSA and SSA

Index	Function value ASSA	Function value SSA	Evaluation Number ASSA	Evaluation Number SSA
1	3.0×10^{-7}	3.2×10^{-4}	5.5×10^{4}	6.9×10^{5}
2	9.3×10^{-7}	1.3×10^{-3}	5.2×10^{4}	7.0×10^{5}
3	5.0×10^{-7}	7.3×10^{-4}	5.0×10^{4}	6.9×10^{5}
4	8.1×10^{-7}	6.1×10^{-4}	5.0×10^{4}	7.0×10^{5}
5	4.9×10^{-7}	1.0×10^{-3}	4.9×10^{4}	7.0×10^{5}
6	4.2×10^{-7}	1.2×10^{-3}	5.4×10^{4}	6.9×10^{5}
7	7.9×10^{-2}	7.8×10^{-4}	5.0×10^{4}	7.2×10^{5}
8	3.7×10^{-7}	8.0×10^{-4}	5.0×10^{4}	7.1×10^{5}
9	6.9×10^{-7}	6.8×10^{-4}	5.0×10^{4}	7.0×10^{5}
10	4.8×10^{-7}	1.4×10^{-3}	4.8×10^{4}	6.9×10^{5}

simulations compared to 6.99×10^5 in SSA. Hence, ASSA is more efficient than SSA. Therefore, ASSA will be used for solving the SSP inversion problems.

5.4.2 SSP inversion

The SSP is assumed to be range independent and the effective SSP for the test case is the idealized Munk profile [27]. The bottom depth is assumed to be 5000m. We only consider inversion of the SSP, hence, we fixed the bottom parameters, such as the thickness of sediment and the sound speed in sediment. A uniform sampling of $c(z)$ in depth at a step of 200m and an additional point at the depth of $250m$ is used for this modeling to give a smooth interpolation of the SSP for the field computation. As the significance of the sound speed variation is positively related to the depth and in deep water, the sound speed tends to be constant in time, we only consider the sound speed of the depths above 800m as the parameters to be estimated. Therefore, there are 6 unknown parameters in this inversion problem. The upper bound and lower bound of the search space are indicated in Fig. 5.2, which is wide at the surface and narrower at deeper depth, represented by dotted lines.

For the acoustic field computation at distances less than 30km, we use the fast field method SCOOTER [24] and for longer distances, we use a normal mode method KRAKEN [100].

In our MFI simulation, the source is set to 50Hz and is fixed at a depth of 200m. The receiver array for both the inversion and the localization is made up of 21 elements which are uniformly placed at a depth range from 10m to 510m.

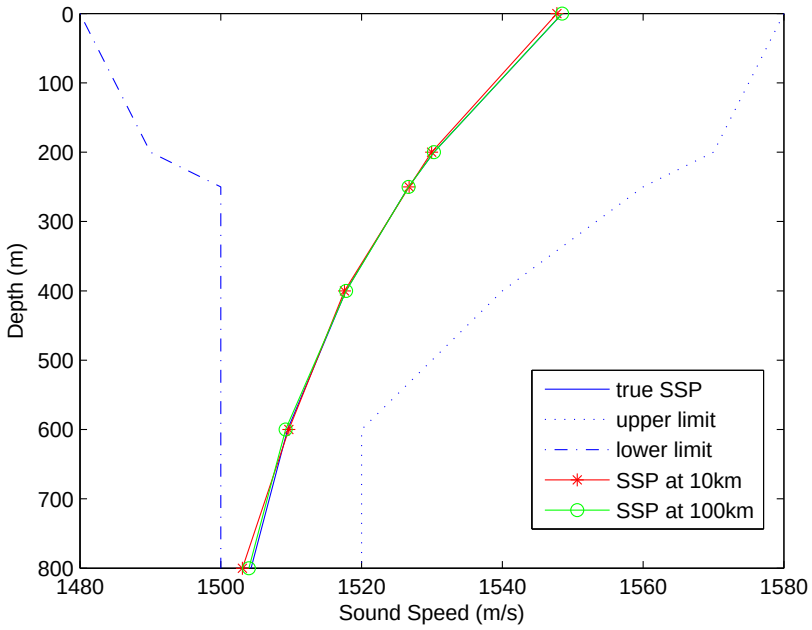


Figure 5.2: Estimated SSP without noise for different distances between source and receiver: 10km and 100km.

The test case has assumed that the position of the acoustic source is known. The quenching is stopped when the tolerance parameter $\epsilon < 10^{-4}$. The search bound for the ith parameter is indicated in Fig 5.2.

Firstly, we consider the case without noise at the measured field so that we can examine the reliability of the global search method. By setting the cooling factor $\beta = 0.99$ and 5 accepted perturbations at each temperature, after 1000 temperature steps, accurate results are achieved in the simulation with mismatch $E < 10^{-5}$. The results for two horizontal distances, which are 10km and 100km are shown in Fig. 5.2, where plots of errors between the true SSP and the estimated SSP as a function of depth are shown. The errors are fairly

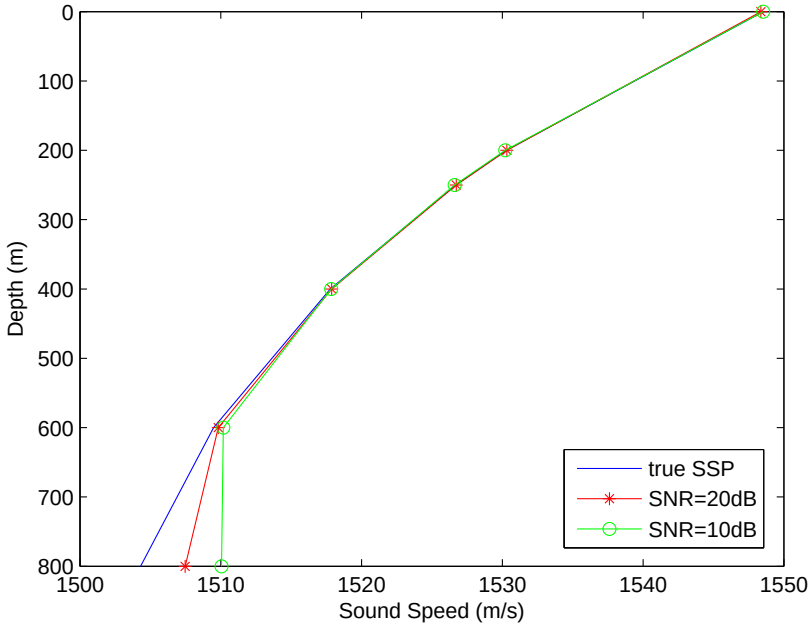


Figure 5.3: Estimated SSP in different noise levels: the distance between source and receiver is 10km.

small with most of the SSP points within ± 0.5m/s interval around the true SSP.

Now we apply this method to a more practical case, in which the noise is added to the measured field. Two SNR levels are tested, 20dB and 10dB. The inversion results are not expected to be as accurate as those without noise, therefore, a faster annealing schedule is used: $\beta = 0.98$ and 500 temperature steps with 3 accepted perturbations. A comparison between the estimation and the true values are shown in Figure 5.3. Compared to the results in noise free cases, the errors between estimated SSP and true SSP are larger. The mismatches are 0.086 for 10dB and 0.004 for 20dB.

To further verify the inversion results, we apply the estimated SSP to source localization problems. Here, we assume a stationary acoustic source and use the estimated SSP from distance of 10km. The ambiguity surface of the the minimum variance distortionless response (MVDR) MF processor [14] is given by:

$$B_{MV} = \frac{1}{\mathbf{w}^H \mathbf{R}^{-1} \mathbf{w}}, \tag{5.22}$$

where $\mathbf{R}$ is the covariance matrix of the received signal and $\mathbf{w}$ is the steering vector (modeled acoustic field generated using known environmental parameters). Locating

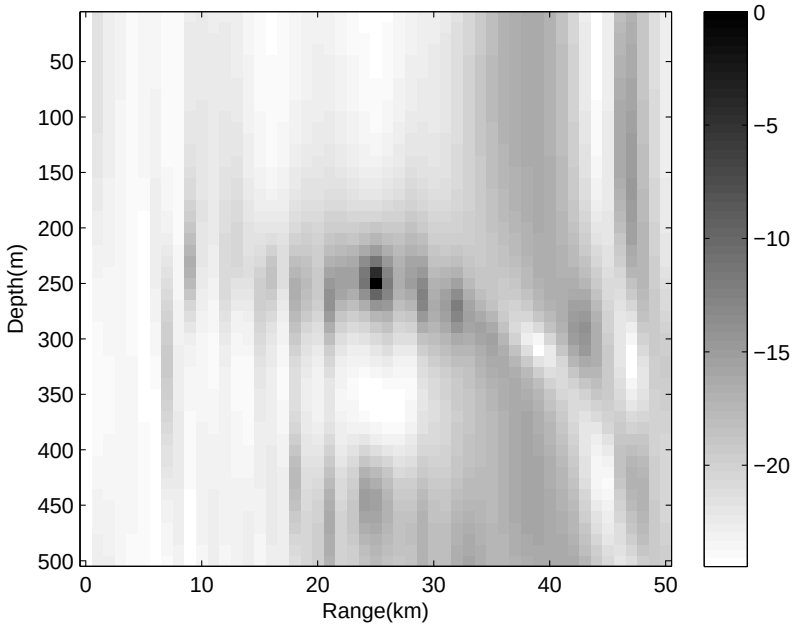


Figure 5.4: Ambiguity surface for source localization: source at the depth of $250m$ and the range of 25km.

an acoustic source in different scenarios (i.e., different source frequencies and different source/receiver positions) have been carried out and accurate localization results have been denmenstrated. Here we only present results of a 100Hz source at a depth of 250m and range of 25km, in which the SSP estimated in the case of SNR $= 10$dB is used. Fig. 5.4 shows the ambiguity surface from the depth of 0m to 500m and the range from 0km to 50km. The stepsize on the range is 1.0km and on depth is 10m. From the results in Fig. 5.4, we can see that the source is located correctly.

5.5 Summary

In this chapter, the MF inversion problems are investigated by using global optimization methods. Global searching algorithms based on simulated annealing are reviewed and compared using simulation results for a multimodal function. The relatively fast global optimization method, adaptive simplex simulated annealing (ASSA) algorithm, is applied to an MF inversion problem for estimating uncertain SSP in a deep water scenario. The inversion results are accurate in the sense that the estimated SSP is close to the true value

and can correctly predict the position of underwater acoustic sources. Positive results in the case of noisy measured field are achieved with a relatively fast convergence speed.

Chapter 6

Estimation of Time-varying Sparse Underwater Acoustic Channels

Contents

Channel estimation is an important task for communication receivers to provide high performance. Many techniques in existing modems rely on the accuracy of channel estimation [9] [101] [102], e.g., such as the channel estimation based linear equalizer and decision feedback equalizer [101]. In this chapter, we investigate estimation of time-varying underwater acoustic channels. One of the main purposes of this chapter is to provide a low-complexity technique for estimating channels with a large number of coefficients. The other one is to exploit the sparsity of underwater acoustic channels to improve the estimation performance, where the sparsity can be estimated according to the source/receiver positions and the acoustic environment.

6.1 Introduction

The underwater acoustic channel is considered as one of the most difficult channels for high speed reliable communications [103]. Due to the low speed of sound, the delay spread of the channel impulse response can be large, on the order from ten to a hundred of symbol durations depending on the specific underwater environment and distance between the transmitter and receiver [7] [8]. For such channels, the number of taps required to represent the impulse response is large. Besides, the channel tap coefficients vary in time. In these cases, an estimator of the channel impulse response would need to track the time variations of a large number of parameters, which is a difficult and computationally expensive task. In many underwater scenarios, the channel impulse response can be sparse [7] [8] [9] [10], e.g., only a small number of taps are non-zeros (with strong tap coefficients). This characteristic makes the estimation of underwater acoustic channels a highly challenging task. Failure in accounting for the sparsity will result in high estimation errors and high computational complexity caused by overparameterization as most of the parameters to be estimated are of little energy. It is beneficial to take into account the channel sparsity by identifying the non-zero channel taps and only estimating the corresponding coefficients for improving the estimation accuracy and reducing the complexity [9]. In this chapter, we investigate low-complexity estimation of time-varying underwater acoustic channels by exploiting the channel sparsity.

BEM based channel estimation methods can be adopted for dealing with time-varying channels [9] [10]. The time variation of each tap coefficient is represented as a linear combination of basis functions. Instead of estimating the time-varying tap coefficients directly, the time-invariant expansion coefficients are estimated. This allows significant reduction in the number of parameters to be estimated [20], thus, resulting in reduced complexity. Among the widely used basis functions are the KL functions [47] [21], DPS functions [20] [22], CE [48] [9] [10], and B-splines [23] [104]. In this work, B-splines are chosen for the high approximation accuracy and low complexity [50] [105]. Specifically, the cubic B-splines [53] are adopted.

Extensive effort has been paid to BEM channel estimation for time-varying channels such as time-varying Rayleigh fading channels [20–23]. The maximum likelihood (ML)

estimation of the expansion coefficients can be obtained directly without adopting the channel statistics [106] [107]. With the statistics of channel time variations, the optimal solution in the sense of minimum mean square error (MMSE) can be achieved [106] [107]. However, for underwater acoustic channels, the statistic information is difficult to obtain, thus, the MMSE solution is hard to achieve. Besides, the large number of taps result in high computational complexity in implementing the ML and the MMSE BEM channel estimator. In order to address these problems in this chapter, we develop an iterative BEM estimator for underwater acoustic channels with a particular concern on the accuracy and the complexity of implementation. Unlike the ML and MMSE BEM methods jointly estimating the expansion coefficients for all the channel taps, the iterative algorithm treats each channel tap separately, thus, allowing a lower complexity.

The rest of this chapter is organized as follows. In Section 6.2, signal model for BEM based channel estimation as well as ML and MMSE BEM estimators are described. The proposed iterative BEM channel estimation method is also presented. In Section 6.3, complexity of the iterative estimator is analyzed. In Section 6.4, a method used for identifying non-zero taps of the channel is described. Numerical results that show the performance of the proposed method and existing ones are provided in Section 6.5. Finally, Section 6.6 gives conclusions.

6.2 Channel estimation based on the basis expansion model (BEM)

6.2.1 Signal model

In a time-varying channel, the discrete-time samples $y(n) = y(nT)$ of the received signal $y(t)$ with a sampling rate $1/T$ can be represented as

$$y(n) = \sum_{k=1}^{K} s(nT - kT)h(nT, kT) + v(nT),\ n = 0, \ldots, N-1, \tag{6.1}$$

where $h(nT, kT)$, $k = 1, \ldots, K$, and $s(n) = s(nT)$ are the channel impulse response and transmitted signal at time nT, respectively; $v(nT)$ is the noise and K is the number of

channel taps. In terms of data-aided channel estimation, the (pilot) signal $s(n)$ is assumed to be known at the receiver.

Stacking N successive received signal samples, (6.1) can be written in the following form

$$\mathbf{y} = \sum_{k=1}^{K} \mathbf{S}_k \mathbf{h}_k + \mathbf{v}, \tag{6.2}$$

where $\mathbf{y} = [y(0), \ldots, y(n), \ldots, y(N-1)]^T$ is an $N \times 1$ vector of the received signal samples, $\mathbf{S}_k$ is an $N \times N$ diagonal matrix with the nth diagonal element $s(nT - kT)$, and $\mathbf{h}_k = [h(0, kT), \ldots, h(nT, kT), \ldots, h((N-1)T, kT)]^T$ is an $N \times 1$ vector representing the variation process of the tap coefficient at delay kT.

Classical adaptive filtering methods such as the least squares method can track slow variations of the channel since the channel is considered as time-invariant for a specific period [10] . However, this assumption may not hold when the channel variations are fast. An alternative method of tracking the channel variations is to use BEM, in which the variations of each tap coefficient are represented as an expansion using basis functions. The corresponding expansion coefficients are time-invariant. Note that this assumption is less restrictive than assuming the channel is constant over an equivalent period of time [10].

In this work, cubic B-spline basis functions [53] are adopted for the channel estimation. Denoting $\mathbf{b}_m = [b(0 - mT_s), \ldots, b(nT - mT_s), \ldots, b((N-1)T - mT_s)]^T$ where $b(nT)$ is the cubic B-spline and T_s the sampling interval separating two adjacent B-spline functions, the fading process of each channel tap can be represented as follows:

$$\mathbf{h}_k = \sum_{m=1}^{M} c_{km} \mathbf{b}_m \tag{6.3}$$

$$= \mathbf{B}\mathbf{c}_k \tag{6.4}$$

where $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_m, \ldots, \mathbf{b}_M]$ is an $N \times M$ basis function matrix and $\mathbf{c}_k = [c_{k1}, \ldots, c_{km}, \ldots, c_{kM}]^T$. The cubic B-spline basis functions (with order $\upsilon = 3$) can be built according to [53]:

$$b(t) = \frac{1}{\upsilon!} \sum_{i=0}^{\upsilon+1} (-1)^i \begin{pmatrix} \upsilon + 1 \\ i \end{pmatrix} \left(\frac{t}{T_s} + \frac{\upsilon + 1}{2} - i \right)^{\upsilon}, \tag{6.5}$$

where $T_s = (N-1)T/(M-\upsilon)$.

The received signal in (6.2) can then be represented using the basis functions as [108]:

$$\mathbf{y} = \sum_{k=1}^{K} \mathbf{S}_k \mathbf{B} \mathbf{c}_k + \mathbf{v}. \tag{6.6}$$

Estimating NK channel coefficients is then converted into a problem of estimating MK expansion coefficients where $M \ll N$.

6.2.2 ML and MMSE BEM channel estimation

An estimate of the expansion coefficients can be obtained by solving the system of equations (6.6) directly. For convenience, (6.6) is rewritten as:

$$\begin{aligned} \mathbf{y} &= \mathbf{S}\mathbf{h} + \mathbf{v}, \\ &= \mathbf{S}\overline{\mathbf{B}}\mathbf{c} + \mathbf{v}, \end{aligned} \tag{6.7}$$

where $\mathbf{S} = [\mathbf{S}_1, \ldots, \mathbf{S}_k, \ldots, \mathbf{S}_K]$ is the $N \times NK$ pilot signal matrix, $\mathbf{h} = [\mathbf{h}_1^T, \ldots, \mathbf{h}_k^T, \ldots, \mathbf{h}_K^T]^T$ is the $NK \times 1$ channel vector, $\mathbf{c} = [\mathbf{c}_1^T, \ldots, \mathbf{c}_k^T, \ldots, \mathbf{c}_K^T]^T$ is the $KM \times 1$ vector containing all the basis expansion coefficients and $\overline{\mathbf{B}} = \mathbf{\Lambda}_K \otimes \mathbf{B}$, where $\mathbf{\Lambda}_K$ is an $K \times K$ identity matrix.

By assuming that the noise $v(t)$ is additive Gaussian, the ML estimate of $\mathbf{c}$ is given by [107]:

$$\hat{\mathbf{c}}_{\text{ML}} = (\overline{\mathbf{B}}^H \mathbf{S}^H \mathbf{S} \overline{\mathbf{B}})^{-1} \overline{\mathbf{B}}^H \mathbf{S}^H \mathbf{y}. \tag{6.8}$$

A more accurate BEM channel estimator, the MMSE BEM estimator, requires the knowledge of the channel statistics and the noise power σ_v^2. For Rayleigh fading channels following Jakes' model [31] [32], the MMSE estimate of $\mathbf{c}$ is given by:

$$\hat{\mathbf{c}}_{\text{MMSE}} = \left[\overline{\mathbf{B}}^H \mathbf{S}^H \mathbf{S} \overline{\mathbf{B}} + \sigma_v^2 \mathbf{R}_c^{-1} \right]^{-1} \overline{\mathbf{B}}^H \mathbf{S}^H \mathbf{y}, \tag{6.9}$$

where $\mathbf{R}_c = E\{\mathbf{c}\mathbf{c}^H\}$ is the covariance matrix of the expansion coefficients, given by [109]:

$$\mathbf{R}_c = (\overline{\mathbf{B}}^H \overline{\mathbf{B}})^{-1} \overline{\mathbf{B}}^H \mathbf{\Upsilon} \overline{\mathbf{B}} (\overline{\mathbf{B}}^H \overline{\mathbf{B}})^{-1}. \tag{6.10}$$

The $N \times N$ matrix $\mathbf{\Upsilon}$ has elements $[\mathbf{\Upsilon}]_{t_1,t_2} = \sigma_h^2 J_0[2\pi f_D(t_1 - t_2)]$, $t_1, t_2 = 1, \ldots, N$. The quantity f_D is the Doppler frequency [109], $J_0(\cdot)$ is the zero-order Bessel function of the first kind and σ_h^2 is the variance of channel coefficients.

In underwater acoustics, different environments may have different channel statistics, depending on the ocean geography, seasons and movements of source/receiver. For instance, considering that underwater acoustic channels with Rayleigh fading are supported by some experimental results [110], while challenged by others [111]. It is usually difficult or expensive to obtain the statistics for underwater acoustic channels, therefore, the MMSE estimate of the channel is hardly achievable. For underwater acoustic channels, there tends to be a large number (K) of taps for representing the impulse response. In the cases that K is large, jointly estimating the expansion coefficients $\mathbf{c}$ for all channel taps, as in (6.8) for the ML estimator, will result in high complexity and require large memory.

6.2.3 Iterative BEM channel estimation

To reduce the complexity of a joint estimation of the expansion coefficients $\mathbf{c}$, we propose an iterative algorithm dealing with each channel tap separately. The idea of the tap-by-tap processing is originated from the ML BEM channel estimator. In (6.8), the matrix $(\overline{\mathbf{B}}^H\mathbf{S}^H\mathbf{S}\overline{\mathbf{B}})$ can be constructed using sub-matrices $(\mathbf{B}^H\mathbf{S}_{k_1}^H\mathbf{S}_{k_2}\mathbf{B})$, $k_1, k_2 = 1, \ldots, K$ as:

$$\overline{\mathbf{B}}^H\mathbf{S}^H\mathbf{S}\overline{\mathbf{B}} = \begin{bmatrix} \mathbf{B}^H\mathbf{S}_1^H\mathbf{S}_1\mathbf{B} & & \cdots & & \mathbf{B}^H\mathbf{S}_1^H\mathbf{S}_K\mathbf{B} \\ & \ddots & & & \\ \vdots & & \mathbf{B}^H\mathbf{S}_{k_1}^H\mathbf{S}_{k_2}\mathbf{B} & & \vdots \\ & & & \ddots & \\ \mathbf{B}^H\mathbf{S}_K^H\mathbf{S}_1\mathbf{B} & & \cdots & & \mathbf{B}^H\mathbf{S}_K^H\mathbf{S}_K\mathbf{B} \end{bmatrix}. \tag{6.11}$$

Compared with the sub-matrices $(\mathbf{B}^H\mathbf{S}_k^H\mathbf{S}_k\mathbf{B})$, since the pseudo-random pilot sequences $s(nT - k_1T)$ and $s(nT - k_2T)$, $k_1 \neq k_2$, $n = 0, \ldots, N-1$, are uncorrelated, the off-diagonal sub-matrices $(\mathbf{B}^H\mathbf{S}_{k_1}^H\mathbf{S}_{k_2}\mathbf{B})$ are small. By ignoring these off-diagonal sub-matrices, a coarse approximation to the ML estimate in (6.8) can be obtained as:

$$\hat{\mathbf{c}}_k = (\mathbf{B}^H\mathbf{S}_k^H\mathbf{S}_k\mathbf{B})^{-1}\mathbf{B}^H\mathbf{S}_k^H\mathbf{y}, \;\; k = 1, \ldots, K. \tag{6.12}$$

Instead of using (6.12) directly, we follow the idea of matching pursuit [112] and adopt an iterative process to build up estimates of the expansion coefficients $\mathbf{c}_k$ gradually. At each iteration, a residual $\mathbf{r}$ is calculated by subtracting a recovered signal from the

received signal $\mathbf{y}$

$$\mathbf{r}^{(q)} = \mathbf{y} - \sum_{k=1}^{K} \mathbf{S}_k \mathbf{B} \hat{\mathbf{c}}_k^{(q-1)} = \mathbf{r}^{(q-1)} - \sum_{k=0}^{K-1} \mathbf{S}_k \mathbf{B} \triangle \hat{\mathbf{c}}_k^{(q)}, \tag{6.13}$$

where q is an index of iterations, $(\cdot)^{(q)}$ denotes the quantity at the qth iteration and $\triangle\hat{\mathbf{c}}_k^{(q)}$, $k = 1, \ldots, K$, are updates of the expansion coefficients, computed as:

$$\triangle\hat{\mathbf{c}}_k^{(q)} = \eta(\mathbf{B}^H \mathbf{S}_k^H \mathbf{S}_k \mathbf{B})^{-1} \mathbf{B}^H \mathbf{S}_k^H \mathbf{r}^{(q)}, \;\; k = 1, \ldots, K, \tag{6.14}$$

where η is a coefficients (step size) in $[0, 1]$. At the initialization stage ($q = 0$), $\mathbf{c}_k^{(0)} = \mathbf{0}$, hence, $\mathbf{r}^{(0)} = \mathbf{y}$. In the following stages ($q \geq 1$), the estimated expansion coefficients $\hat{\mathbf{c}}_k$ are updated according to:

$$\hat{\mathbf{c}}_k^{(q)} = \hat{\mathbf{c}}_k^{(q-1)} + \triangle\hat{\mathbf{c}}_k^{(q)}, \;\; k = 1, \ldots, K. \tag{6.15}$$

At the start of the iterations, the residual is high. By extracting estimates of the expansion coefficients and subtracting recovered signal from the residual, the residual is gradually reduced. During this process, the estimates $\hat{\mathbf{c}}_k$ are improved and finally converge to a steady state.

For each iteration, (6.14) is repeated for all the K channel taps. For the potentially large K and total number of iterations Q, computing the matrix $(\mathbf{B}^H \mathbf{S}_k^H \mathbf{S}_k \mathbf{B})^{-1}$ can result in high complexity. To reduce the complexity, the following simplified formulation is found useful:

$$\triangle\hat{\mathbf{c}}_k^{(q)} = \eta(\mathbf{B}^H \mathbf{B})^{-1} \mathbf{B}^H \mathbf{S}_k^H \mathbf{r}^{(q)}, \;\; k = 1, \ldots, K. \tag{6.16}$$

In this case, the matrix $(\mathbf{B}^H \mathbf{B})^{-1}$ can be pre-computed and kept in memory to reduce the complexity. For modulations such as the phase-shift keying where the modulated symbols have a unit magnitude $|s(n)| = 1$, (6.16) is equivalent to (6.14). For other modulations that have a mean symbol power $E\{|s(n)|^2\} = 1$, (6.16) is an approximation to (6.14).

It is found that the diagonal elements of $\mathbf{B}^H \mathbf{B}$ are much larger than the off-diagonal elements, therefore, (6.16) can be further simplified as:

$$\triangle\hat{\mathbf{c}}_k^{(q)} = \xi \mathbf{B}^H \mathbf{S}_k^H \mathbf{r}^{(q)}, \tag{6.17}$$

where $\xi = \eta / \max\{\mathbf{B}^H \mathbf{B}\}$ is a normalization factor. The choice of η will influence the final estimation accuracy and convergence speed in terms of number of iterations. We will show how to choose η according to numerical results in Section 6.5.

The iterations stop when the desired number of iterations Q is reached. The final estimate of the channel impulse response is then calculated as:

$$\hat{\mathbf{h}} = \overline{\mathbf{B}}\hat{\mathbf{c}}^{(Q)}. \tag{6.18}$$

The whole iterative algorithm is summarized in Table 6.1.

6.3 Complexity analysis

So far, we have described the iterative BEM channel estimator. We now compare its computational complexity with that of the ML BEM estimator (the MMSE BEM estimator has similar complexity with the ML BEM estimator). To start with, we compute the complexity of the ML BEM estimator. The key steps and the corresponding complexity, in terms of number of multiplications and additions, are shown in Table 6.2. At step 1, matrices $\mathbf{S}_k\mathbf{B}$, $k = 1, \ldots, K$, are computed. As $\mathbf{S}_k$ is a diagonal matrix, the computation of $\mathbf{S}_k\mathbf{B}$ only involves KMN multiplications. By noticing that the cubic B-spline function has a finite support, i.e., each row of matrix $\mathbf{B}$ has maximum 4 non-zero elements, the computation of $\mathbf{S}_k\mathbf{B}$ requires $4KN$ multiplications. The results of step 1 are used for computing $\mathbf{B}^H\mathbf{S}_k^H\mathbf{y}$ at step 2 and constructing $\overline{\mathbf{B}}^H\mathbf{S}^H\mathbf{S}\overline{\mathbf{B}}$ at step 3. Different methods can be applied for computing the matrix inversion at step 4. Generally, the inversion of the $KM \times KM$ matrix $\overline{\mathbf{B}}^H\mathbf{S}^H\mathbf{S}\overline{\mathbf{B}}$ requires $\mathcal{O}(K^3M^3)$ multiplications and additions. By summing the number of operations at each step and ignoring the terms with low order, the total complexity of the ML BEM estimator is $\mathcal{O}\left(8K^2N\right) + \mathcal{O}(K^3M^3)$.

For the iterative BEM channel estimator, there are four key steps at each iteration, as shown in Table 6.1. The matrices $\mathbf{S}_k\mathbf{B}$, $k = 1, \ldots, K$, are pre-computed and kept in memory to reduce the complexity at step 1 and step 3. The total complexity of the iterative BEM estimator is $\mathcal{O}\left(8QKN\right)$.

From Table 6.1 and Table 6.2, it can be seen that the relationship between the complexity of the ML estimator and that of the iterative estimator depends on values of K and Q. Generally, when the number of channel taps K is large, the iterative estimator has lower complexity. As an example, for the simulation scenarios used in Section 6.5.2, the underwater acoustic channel has the number of taps $K \sim 10^2$. The block size N is

Table 6.1: Iterative BEM channel estimator using cubic B-splines

	Input: M, N, $\mathbf{y}$, $\mathbf{B}$, η, Q, $\mathbf{S}_k$, $k = 1, \ldots, K$	$\times$	$+$
Step	**Initialization**: $\xi = \eta / \max\{\mathbf{B}^H\mathbf{B}\}$ **repeat for K times** $\hat{\mathbf{c}}_k^{(0)} = \xi\mathbf{B}^H\mathbf{S}_k^H\mathbf{y}$ **end for** $\mathbf{r}^{(0)} = \mathbf{y} - \sum_{k=1}^{K}\mathbf{S}_k\mathbf{B}\hat{\mathbf{c}}_k^{(0)}$ Compute residual power: $P_r = \|\|\mathbf{r}^{(0)}\|\|_2^2/N$ $q = 1$		
	Iteration: **While** $q \leq Q$		
1	**repeat for K times** $\triangle\mathbf{c}_k^{(q)} = \xi\mathbf{B}^H\mathbf{S}_k^H\mathbf{r}^{(q-1)}$	$(4K+1)N$	$K(4N-M)$
2	$\hat{\mathbf{c}}_k^{(q)} = \hat{\mathbf{c}}_k^{(q-1)} + \triangle\mathbf{c}_k^{(q)}$ **end for**	$-$	KM
	$q = q + 1$		
3	$\mathbf{r}^{(q)} = \mathbf{y} - \sum_{k=1}^{K}\mathbf{S}_k\mathbf{B}\hat{\mathbf{c}}_k^{(q-1)}$	$4KN$	$K(4N-M)+N$
4	Compute: $P_r = \|\|\mathbf{r}^{(q)}\|\|_2^2/N$	$N+1$	$N-1$
	end While		
	Final channel estimate: $\hat{\mathbf{h}} = \overline{\mathbf{B}}\hat{\mathbf{c}}^{(Q)}$	Total: $\mathcal{O}(8QKN)$ multiplications and additions	

on the order of 10^2, the number of basis function is about 10, and the maximum number of iterations Q is about 20. The iterative estimator requires $\mathcal{O}(10^6)$ operations, while the ML estimator requires $\mathcal{O}(10^9)$ operations.

6.4 Non-zero taps identification

Underwater acoustic channels may be sparse [7] [8] [9] [10], e.g., only a small number of tap coefficients are non-zeros. The channel impulse response can be represented by only the non-zero taps. Estimating the coefficients of the zero-taps (with very small tap coefficients) results in overparameterization, which in turn results in low accuracy and high estimation complexity. Therefore, it is beneficial to identify the non-zero taps before applying the proposed iterative BEM channel estimator. Since the iterative estimator involves a tap-by-tap processing, it can be straightly applied by only processing the non-

Table 6.2: Complexity of ML BEM channel estimator using cubic B-splines

Step	Operation	$\times$	$+$
1	Compute: $\mathbf{S}_k\mathbf{B}$, $k = 1, \ldots, K$	$4KN$	$-$
2	Compute: $\mathbf{B}^H\mathbf{S}_k^H\mathbf{y}$, $k = 1, \ldots, K$	$4KN$	$K(4N - M)$
3	Construct: $\overline{\mathbf{B}}^H\mathbf{S}^H\mathbf{S}\overline{\mathbf{B}}$	$\frac{16K(K-1)N}{2}$	$\frac{16K(K-1)(N-1)}{2}$
4	Compute: $(\overline{\mathbf{B}}^H\mathbf{S}^H\mathbf{S}\overline{\mathbf{B}})^{-1}$	$\mathcal{O}(K^3M^3)$	$\mathcal{O}(K^3M^3)$
5	Compute: $\hat{\mathbf{c}}_{\text{ML}} = (\overline{\mathbf{B}}^H\mathbf{S}^H\mathbf{S}\overline{\mathbf{B}})^{-1}\overline{\mathbf{B}}^H\mathbf{S}^H\mathbf{y}$	K^2M^2	$KM(KM-1)$
	Total: $\mathcal{O}\left(8K^2N\right) + \mathcal{O}(K^3M^3)$ multiplications and additions		

zero taps. Consequently, the estimation complexity is reduced since the total number of taps K is reduced. In this section, a method for identifying the non-zero taps is described.

Various methods can be used for identifying the non-zero taps. A straightforward method is the threshold algorithm [101] [113], where the dominant taps can be identified using an energy criteria: the non-zero taps should have coefficients stronger than a specific threshold. Another type of methods solving this problem is to find a reduced number of taps that can represent the whole channel impulse response with small error [10]. Among these methods are the L_p-norm regularization method [74] and greedy algorithms such as the matching pursuit [112].

In this work, we identify the non-zero channel taps by solving a BPDN problem, which is a l_1 norm regularized optimization problem:

$$\min_{\hat{\mathbf{c}}} J[\hat{\mathbf{c}}], \tag{6.19}$$

where

$$J[\hat{\mathbf{c}}] = \frac{1}{2}\left|\left|\mathbf{y} - \sum_{k=1}^{K}\mathbf{S}_k\mathbf{B}\hat{\mathbf{c}}_\mathbf{k}\right|\right|_2^2 + \tau\left|\left|\hat{\mathbf{c}}\right|\right|_1. \tag{6.20}$$

The parameter τ controls the balance between the square error (l_2 term) and the sparsity (l_1 term). The purpose is to find the indexes of the non-zero taps. We denote the non-zero indexes as a vector $\mathbf{I} = [I_1, I_2, \ldots, I_{N_n}]$, where N_n is the number of non-zero taps. To solve this BPDN problem, the Homotopy approach [114] developed in Chapter 4 is adopted. The key idea of the Homotopy approach is that for a large value of τ, all the taps $\hat{\mathbf{c}}_k$, $k = 1, \ldots, K$ should be zeros. When τ is reduced gradually, e.g., $\tau \leftarrow \lambda\tau$, $0 < \lambda < 1$, non-zero taps are activated to minimize the cost function in (6.20).

It has been proved in Chapter 4 that activating the nth non-zero tap (with index t) to

minimize the cost function for a specific τ, the following criteria should be satisfied:

$$[\overline{\mathbf{g}}]_t = \max\{\mathbf{g}_t\} > \tau, \tag{6.21}$$

where $\mathbf{g}_t = \left|\mathbf{B}^H \mathbf{S}_k^H \mathbf{r}\right|$. The vector $\mathbf{r} = \mathbf{y} - \sum_{i=1}^{n-1} \mathbf{S}_{I_i} \hat{\mathbf{c}}_i$ is the current residual, and $\overline{\mathbf{g}}$ is an auxiliary vector with elements $[\overline{\mathbf{g}}]_k = \max\{\mathbf{g}_k\}$, $k = 1, \ldots, K$.

At the initialization ($\hat{\mathbf{c}} = \mathbf{0}$), the parameter τ is set to $\tau = \max\{\overline{\mathbf{g}}\}$ so that the first non-zero tap has an index $t = \arg\max_k\{\overline{\mathbf{g}}\}$. The CD search is used to adjust the coefficients for this non-zero tap to minimize the cost function. When τ is reduced, more non-zero taps can be activated. We choose each new non-zero tap that has the index $t = \arg\max_{k \in \bar{\mathbf{I}}}\{\overline{\mathbf{g}}\}$, where $\bar{\mathbf{I}}$ is the complementary set of the indexes in vector $\mathbf{I}$. The Homotopy process is terminated when the desired number of non-zeros taps (N_n) is reached. The algorithm is summarized in Table 6.3.

6.5 Numerical results

In this section, the proposed iterative channel estimation method is firstly evaluated using non-sparse channels with short delay spread, comparing with the ML and MMSE BEM channel estimation methods. The proposed method is then applied to estimate a time-varying sparse underwater acoustic channel with relatively large delay spread.

6.5.1 Numerical results for non-sparse short channels

We consider a time-varying multipath channel that has $K = 5$ taps. The tap coefficients are assumed independent from each other. A uniform power delay profile is used, e.g., each tap of the channel has the same energy. The complex-valued time-varying tap coefficients are generated according to a modified Jakes' model [31] [32]. Throughout the simulations in this subsection, the number of received symbols is fixed at $N = 200$. A binary pseudo-random pilot signal sequence is used for the channel estimation. The MSE of the channel estimate is averaged over 100 simulation trials, where the MSE is calculated

Table 6.3: Algorithm for identifying non-zero channel taps

Input:

N_n, λ, $\mathbf{B}$, $\mathbf{y}$, S_k, $k = 1, \ldots, K$

Initialization:

$\hat{\mathbf{c}} = \mathbf{0}$, $\mathbf{r} = \mathbf{y}$

Compute for K channel taps:

$\mathbf{g}_k = \left|\mathbf{B}^H \mathbf{S}_k^H \mathbf{r}\right|$, $k = 1, \ldots, K$

Formulate vector $\overline{\mathbf{g}}$, $[\overline{\mathbf{g}}]_k = \max\{\mathbf{g}_k\}$

$t = \arg\max_k\{\overline{\mathbf{g}}\}$

Activate the first non-zero tap:

$I_1 = t$, $\tau = [\overline{\mathbf{g}}]_t$, $m = 1$

While $m \leq N_n$

for $i = I_1, \ldots, I_n$

Adjust $\hat{\mathbf{c}}_i$ to minimize the cost function $J[\hat{\mathbf{c}}]$ using CD search

Update the residual signal: $\mathbf{r} = \mathbf{r} - \mathbf{S}_i \hat{\mathbf{c}}_i$

end for

Update the vectors:

$\mathbf{g}_k = \left|\mathbf{B}^H \mathbf{S}_k^H \mathbf{r}\right|$, $k = 0, \ldots, K-1$

Formulate vector $\overline{\mathbf{g}}$, $[\overline{\mathbf{g}}]_k = \max\{\mathbf{g}_k\}$

$t = \arg\max_{k \in \bar{\mathbf{I}}}\{\overline{\mathbf{g}}\}$

if $[\overline{\mathbf{g}}]_t > \tau$

Update the index of non-zero taps:

$I_i = I_{i-1}$, $i = 2, \ldots, n+1$

$I_1 = t$

$m = m + 1$

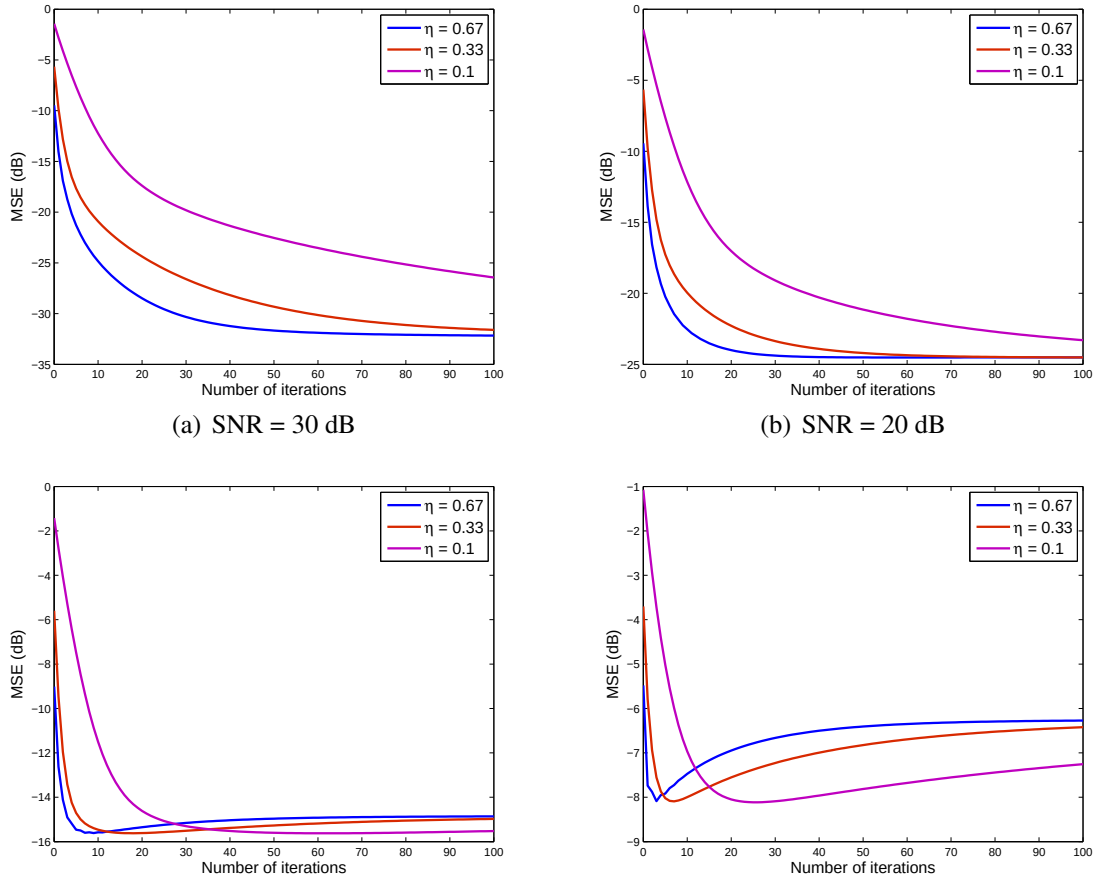
else

$\tau = \lambda\tau$

end if

Output:

Non-zero index $\mathbf{I} = [I_1, I_2, \ldots, I_{N_n}]$



(a) SNR = 30 dB (b) SNR = 20 dB

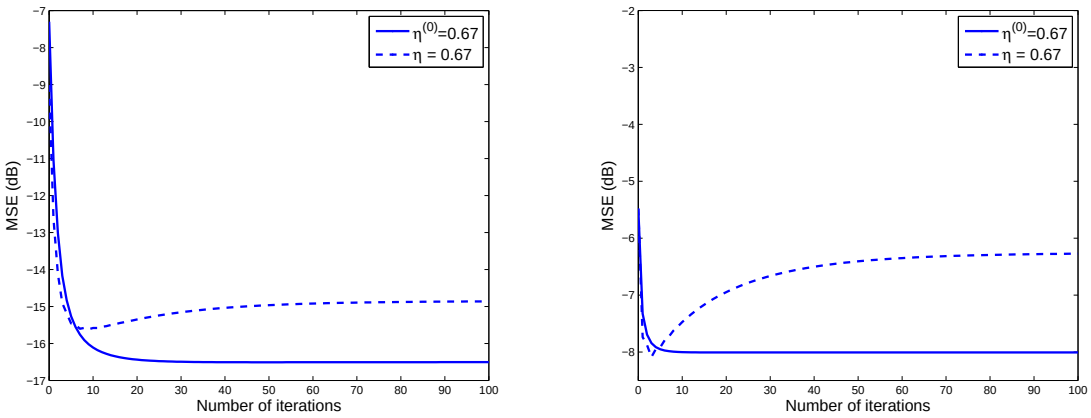
(c) SNR = 10 dB (d) SNR = 0 dB

Figure 6.1: MSE performance of the iterative BEM channel estimator with respect to the number of iterations (q): η is constant over the iterations, Non-sparse fading channels with $f_dT = 0.01$, $M = 10$.

as:

$$\text{MSE} = \frac{\sum_{k=1}^{K} ||\hat{\mathbf{h}}_k - \mathbf{h}_k||_2^2}{\sum_{k=1}^{K} ||\mathbf{h}_k||_2^2}. \tag{6.22}$$

The MSE of channel estimates should decrease when the number of iterations increases. Fig.6.1 shows the convergence of the iterative estimator for different values of η. In the simulation, each tap of the channel is assumed to have a normalized Doppler frequency $f_dT = 0.01$. The number of basis functions is fixed at $M = 10$. It can be seen that for different values of η, the convergence speeds are different. For high SNR scenarios corresponding to Fig.6.1(a)(b), setting η to a higher value requires less iterations for the MSE converging to a steady state. For lower SNR scenarios corresponding to Fig.6.1(c)(d), there is an optimal number of iterations providing the lowest MSE. Beyond this optimal number, the MSE increases, resulting in degradation in the estimation



(a) SNR = 10 dB (b) SNR = 0 dB

Figure 6.2: MSE performance of the iterative BEM channel estimator with respect to the number of iterations (q) for: a constant η and a varying η ($\eta^{(q)} = \frac{\eta^{(q-1)}}{1+1/\text{SNR}}$, $1 < q \leq Q$), Non-sparse fading channels with $f_d T = 0.01$, $M = 10$.

performance.

From the results shown in Fig.6.1(c)(d), reducing the value of η can avoid or alleviate the degradation in low SNR scenarios. Therefore, instead of setting a constant η for all the iterations, we can set η varying for different iterations. The parameter η should decay as more iterations are carried out. This is because the SNR of the residual decreases as the number of iterations increases. The higher the noise level, the faster this decaying should be. In this work, the following decaying process for the parameter η is found to be useful to alleviate the degradation in low SNR scenarios:

$$\eta^{(q)} = \frac{\eta^{(q-1)}}{1 + 1/\text{SNR}}, \quad 1 < q \leq Q. \tag{6.23}$$

Fig.6.2 shows the MSE performance of the iterative estimator using a decaying η with $\eta^{(0)} = 0.67$. With the variable step-size, a better MSE performance is achieved than that when using a constant η. The simulation scenarios in Fig.6.2 are the same as those for Fig.6.1.

Fig.6.3 shows the MSE performance against SNR in the same simulation scenario as that for Fig.6.1 and Fig.6.2. The MSE performance obtained by applying a constant $\eta = 0.67$ and a varying η with $\eta^{(0)} = 0.67$ are compared with the performance of the ML and MMSE BEM channel estimators. It can be seen that the varying step-size η provides

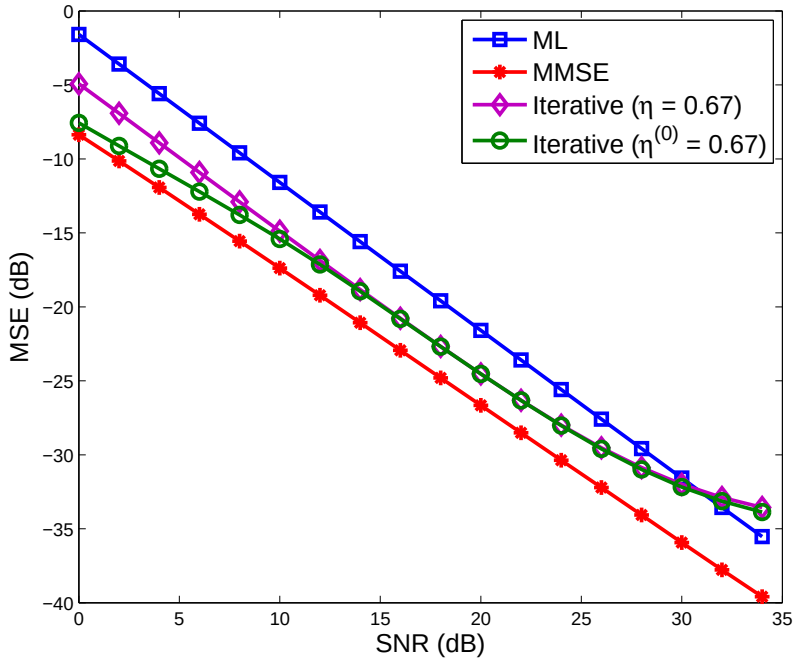


Figure 6.3: MSE of channel estimates at different SNRs: Non-sparse fading channels with $f_dT = 0.01$, $M = 10$, $Q = 50$, $N = 200$.

a better estimation accuracy for low SNRs than the constant η. The iterative estimator, with either a constant or varying η, can provide more accurate channel estimation than the ML BEM estimator for SNR < 30 dB. Since in real communication systems, very high SNR (> 25 dB) is difficult and sometimes unnecessary to achieve, we can conclude that the proposed estimator has a better performance than the ML BEM estimator.

We also investigate the influence of the number of basis functions (M) on the channel estimation accuracy for the proposed iterative algorithm. Two different fading rates are considered: $f_dT = 0.01$ and $f_dT = 0.02$. Fig.6.4 shows the MSE performance with respect to the number of basis functions. To justify the proposed method, the performance of the ML and MMSE BEM methods are also shown. It can be seen that for the MMSE BEM estimator, there exists an optimal number of basis functions above which a lower bound of the MSE performance is achieved. The behavior of the MSE performance of the ML BEM estimator is different from that of the MMSE estimator. There is an optimal number of basis functions providing the best MSE performance. Fewer or higher number of basis functions will produce less accurate estimates. Similar to the ML estimator, the proposed estimator also has an optimal number of basis functions. However, beyond this

number, the performance is less sensitive to the increase of the number of basis functions.

6.5.2 Numerical results for modeled underwater acoustic channels

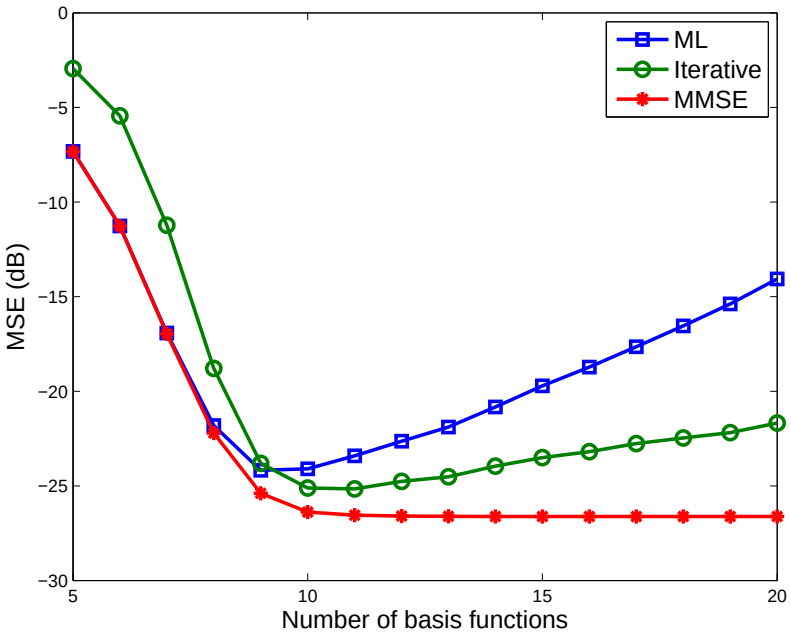
The time-varying underwater acoustic channel is generated according to the environment of a deep water experiment conducted in the Pacific Ocean [44]. More details about the environment parameters such as the SSP can be found in Chapter 3. In the simulation, we follow the settings of source/receiver movements in the experiment. The transmitter, at a depth of 200 m, is moving towards the receiver from a horizontal distance of about $r_0 = 42$ km. The receiver depth is fixed at 400 m. The moving speed is time-varying, with a mean speed of $v_0 = 6$ m/s plus sinusoidal fluctuations. The distance between the transmitter and receiver is defined by:

$$r(t) = r_o - v_0 t + \mu \sin(2\pi t/T_{sur}), \tag{6.24}$$

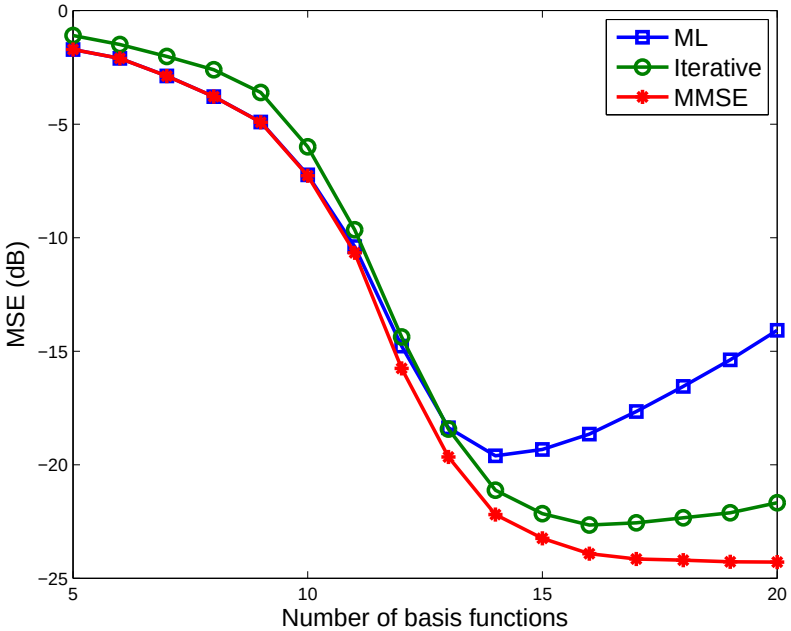
where $\mu = 1$ m and $T_{sur} = 10$ s is the average period of the surface waves. The time-varying acoustic channel is generated using the channel model presented in Chapter 3.

At the transmitter end, the transmitted QPSK symbols $s(n)$ are firstly up-sampled to match the sampling frequency at $f_s = 12288$ Hz and then passed through a root-raised cosine filter with a roll-off factor of 0.5. The symbol duration is $T = 1/1024$ s. The output of the filter is modulated to a carrier frequency of $f_c = 3072$ Hz to produce the real-valued transmitted signal. The channel impulse response is generated in a frequency bandwidth $[1500, 4600]$ Hz to cover the signal bandwidth. For a movement of duration 80 s, the variations of the magnitude of the time-varying impulse response in the bandwidth $[1500, 4600]$ Hz are shown in Fig.6.5, where the time-varying delays of the multipath arrivals caused by the movement are compensated using the delay adjustment method as described in Chapter 3.

We carry out two simulations to examine the proposed iterative BEM estimator with the non-zero tap identification method. In the first simulation, the transmitted symbols $s(n)$ are treated as a pilot known to the receiver. To exclude the Doppler effect caused by the time-varying delays, the time-varying impulse response (after delay adjusement) shown in Fig.6.5 is used for generating the simulated signal. At the receiver end, the



(a) Non-sparse fading channels with $f_dT = 0.01$, $N = 200$, SNR=20 dB, $\eta^{(0)} = 0.67$, $Q = 50$



(b) Non-sparse fading channels with $f_dT = 0.02$, $N = 200$, SNR=20 dB, $\eta^{(0)} = 0.67$, $Q = 50$

Figure 6.4: MSE of channel estimates for three different BEM methods with respect to the number of basis functions.

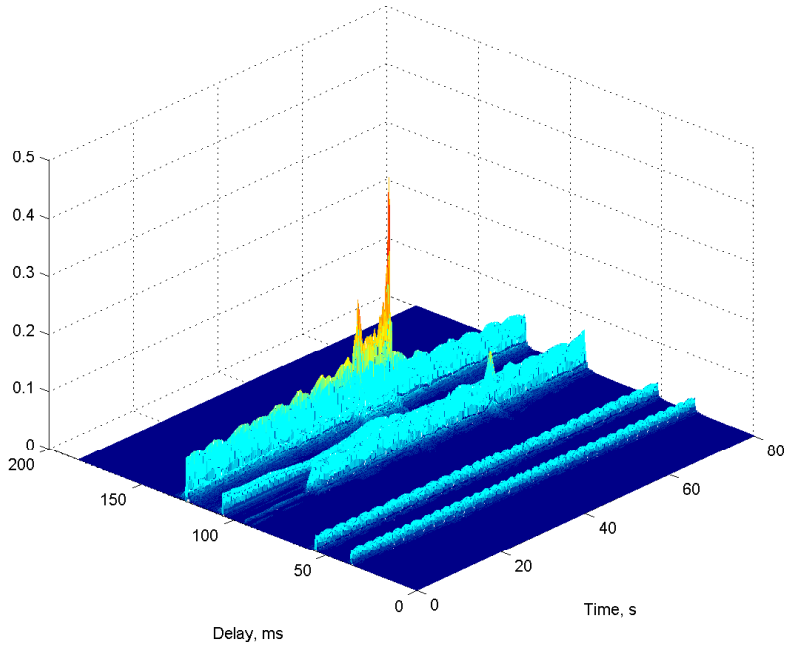


Figure 6.5: Magnitude of the channel impulse response variations in a deep water scenario caused by a horizontal movement.

received signal is firstly converted to its equivalent baseband by removing the carrier frequency and passing through a low-pass filter matched to the transmit filter. The complex-valued received symbols $y(n)$ are used for channel estimation. The signal is processed block by block. The block has a duration of 5 s, with $N = 5120$ symbols. The number of basis functions in a block is fixed at $M = 13$.

It can be seen from Fig.6.5 that the channel is sparse, e.g., there are only several groups of strong channel taps. The channel impulse response can be approximately represented by a small number of non-zero taps. The channel sparsity, defined as the ratio between the number of non-zero taps N_n and the total number of taps K (i.e., N_n/K), can be estimated according to the source/receiver positions and the environment. For instance, with approximate environmental parameters and positions, it can be calculated that there are five groups of multipath arrivals, with a delay spread of about 0.25 s ($K \approx 256$ channel taps). For signals with a limited bandwidth, a distinct path will have some sidelobes. The wider the bandwidth, the narrower these sidelobes will be. If we ignore all the sidelobes and represent each group of the multipath arrivals by one tap, the sparsity is about $5/256 \approx 0.02$. To achieve a better approximation, we can increase the number of samples

for representing each group of the multipath arrivals. For instance, with 3 taps per group, the sparsity is about $5 \times 3/256 \approx 0.06$.

For estimating a sparse channel, two parts of the estimation error should be considered. The first part is the modeling error caused by the sparse approximation for bandlimited signal. Generally, establishing the sparse model with more taps results in less modeling error. The second part of the estimation error is due to the noise in the received signal. For a strong channel tap, this error is small as the SNR for this tap is high. However, for weak taps, this error can be high, especially when the noise level is high. Fig.6.6 shows the MSE performance with respect to the sparsity for different SNRs. For SNR $= 0$ dB, the estimation error caused by noise dominants in the overall estimation error when the sparsity is larger than 0.04. With higher SNR (SNR $= 10$ dB), more taps are required to achieve the optimal estimation accuracy. The performance loss by introducing more than the optimal number of taps is smaller than that at low SNR (SNR $= 0$ dB).

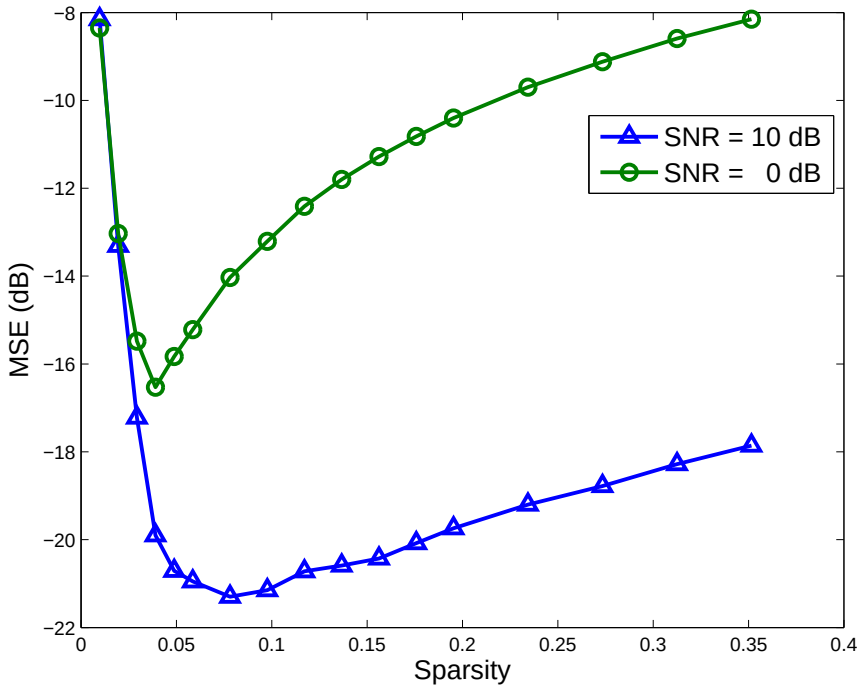


Figure 6.6: MSE of channel estimates obtained by the iterative algorithm for underwater acoustic channel with respect to the sparsity.

Fig.6.7 shows sparse and non-sparse channel estimates for two different SNRs. It is clear that for SNR $= 30$ dB, there are non-zero taps that are not taken into account by

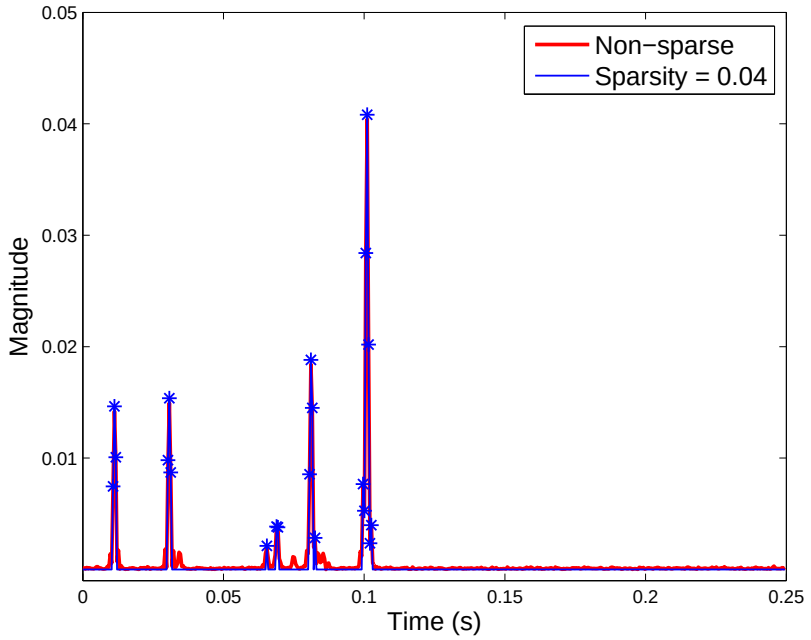
setting the sparsity to 0.04. In contrast, for SNR $= 0$ dB, since the noise level is high, these taps are hardly visible in the non-sparse estimate.

Fig.6.8 compares the MSE performance of the iterative BEM estimator at different SNRs with and without accounting for the channel sparsity. It is seen that for high SNR, the non-sparse estimator provides better performance. However, for lower SNRs (< 15 dB), properly choosing the sparsity results in significant improvement in the channel estimation accuracy.

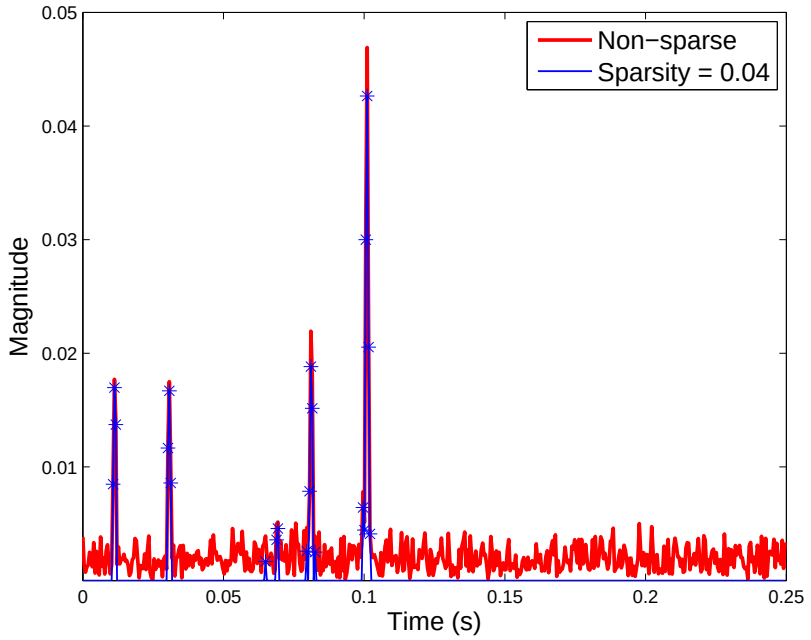
So far, we have examined the proposed methods for estimating time-varying sparse channels with pilot-only signal in a scenario where the time-varying delays are compensated. Next, we carry out a more realistic simulation to evaluate the performance of the channel estimator. In this simulation, the time-varying delays of the multipath arrivals are included when generating the signal. The time-varying impulse response is generated according to the movement in (6.24). It is worth to note that simulated signals generated from these simulation settings are similar to the experimental signals, i.e., they provide close system performance in an OFDM modem, as verified in Chapter 3. The real part of the QPSK symbols is treated as the pilot signal while the imaginary part contains the data unknown to the receiver. The SNR is defined as the ratio between the power of the signal, including the pilot and the data, and the power of the noise.

A block diagram of the receiver is shown in Fig.6.9. At the beginning of processing the received signal, the time-varying Doppler effect caused by the movement is compensated by re-sampling the signal according to an estimated Doppler shift. To simplify the compensation, it is assumed that the Doppler shift is constant within a specific interval T_d. Different intervals of data can have different Doppler shifts. The ambiguity function method [11] is used to estimate the Doppler shift for each interval. The Doppler estimation interval is set to $T_d = 1$ s. The signal after re-sampling is then converted into its equivalent baseband $y(n)$ by removing the carrier frequency, passing through a low-pass filter and down-sampling to the symbol rate. The channel estimates $\hat{\mathbf{h}}$ are used to compute the coefficients of symbol-spaced decision feedback equalizer, with feedforward length $L_f = 2K$ and feedback length $L_b = K$.

For channel estimation, the data are treated as noise. Therefore, the effective SNR for



(a) SNR=30 dB



(b) SNR=0 dB

Figure 6.7: Sparse and non-sparse channel estimates obtained by the iterative algorithm at different SNRs: the sparse taps are marked with '*'.

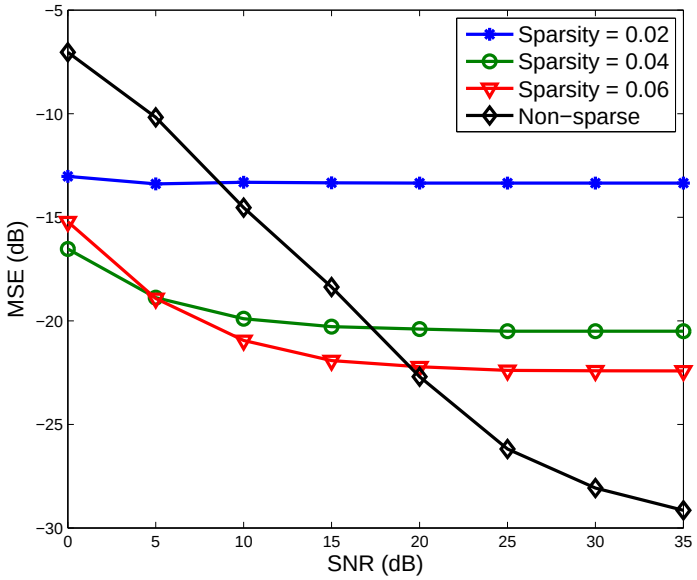


Figure 6.8: MSE of channel estimates obtained by the iterative algorithm for underwater acoustic channel at different SNRs: sparse and non-sparse iterative BEM solutions.

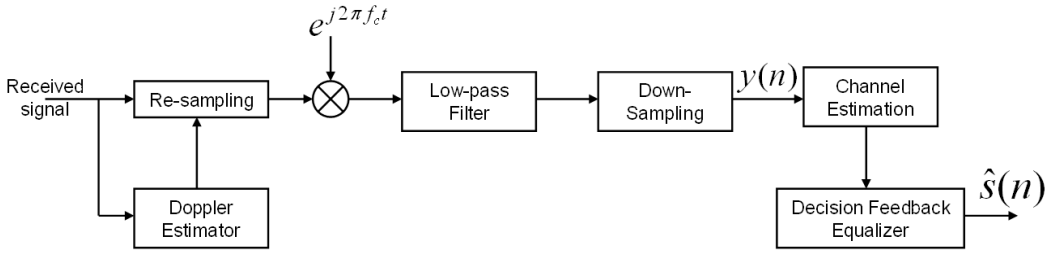


Figure 6.9: Block diagram of the receiver.

channel estimation is always lower than 0 dB. According to the results in Fig.6.8, the non-sparse estimation will result in large error. This is verified again by the results shown in Fig.6.10 and Fig.6.11, where the BER and MSE performance of the non-sparse estimation are poor. In comparison, the sparse channel estimates provide much better BER and MSE performance.

6.6 Summary

In this chapter, we have proposed an iterative BEM based channel estimator using local splines for estimating channels with large number of taps. The iterative estimator involves

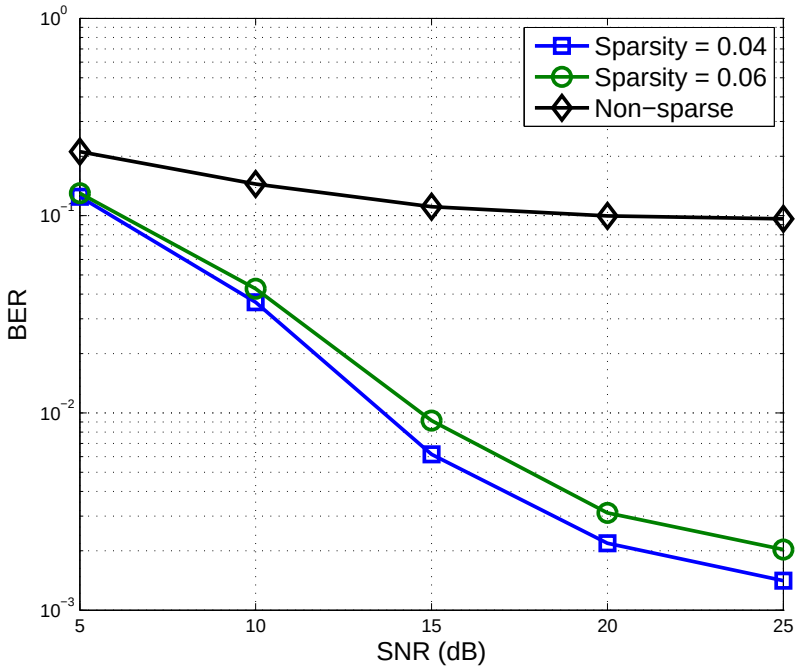


Figure 6.10: BER versus SNR for signals with pilot and data transmitted through the underwater acoustic channel.

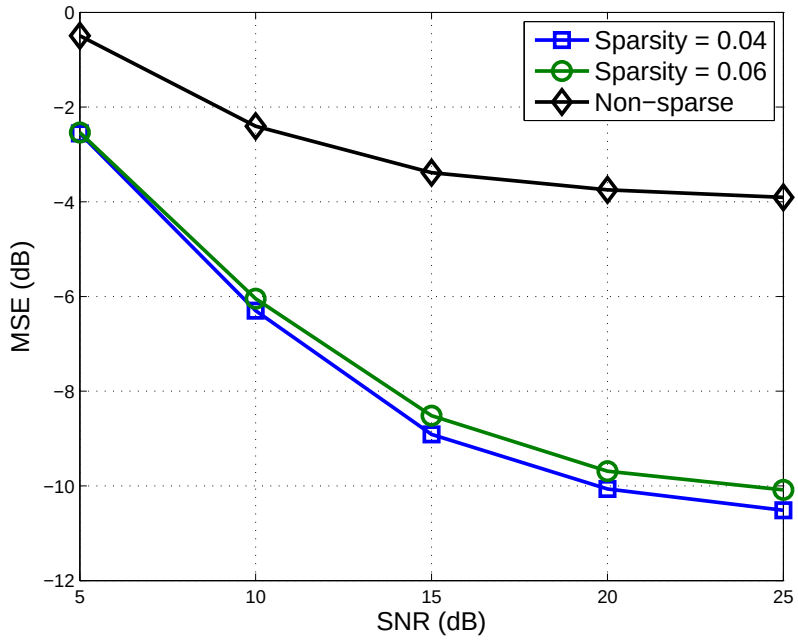


Figure 6.11: MSE versus SNR for signals with pilot and data transmitted through the underwater acoustic channel.

a tap-by-tap processing, dealing with each channel tap separately. It has been shown that for estimating channels with a large number of taps, the proposed iterative estimator has a lower complexity than the ML and MMSE BEM channel estimators. We have shown using Rayleigh fading channels following Jakes' model that the proposed estimator has a better estimation accuracy than the ML BEM estimator. We have also investigated the influence of sparsity of underwater acoustic channels on the performance of channel estimation. The non-zero channel taps are identified by solving the BPDN problem using a Homotopy approach. The proposed iterative channel estimation method and the non-zero tap identification algorithm are examined by simulations with a modeled time-varying underwater acoustic channel. It has been verified that by properly taking the channel sparsity into account, more accurate channel estimate and better system performance can be achieved.

Chapter 7

Summary and Future Work

Contents

7.1 Summary

In this thesis, several signal processing techniques for underwater acoustic communication networks have been developed and investigated.

In Chapter 3, an efficient channel simulator for modeling signal transmission with moving transmitter/receiver has been developed to provide a simulation tool for research in underwater acoustics. The key idea is to generate a time-varying channel impulse response according to the time-varying transmitter/receiver position. The complexity is kept low by approximating the time-varying impulse response at signal sampling rate using local splines over a set of impulse responses for a relatively small number of sampling points (waymarks) on the transmitter/receiver trajectory. The waymark impulse responses are computed using an acoustic field computation method, which are usually of high complexity. Further reduction in the complexity has been achieved by developing a delay adjustment method to reduce the sampling rate of the trajectory. The simulator

can be implemented in a recursive form and, thus, can be used for developing real-time simulators for long-duration communication sessions. The simulator has been verified by comparing the simulated data with data from real ocean experiments in shallow and deep water. For the shallow water experiment, we have shown that the Doppler spectrum of the tones transmitted by a moving source is similar in the simulation and experiment. For the deep water experiment, it has been shown that the performance of an OFDM receiver is similar in the simulation and experiment.

Considering the importance of the knowledge of node positions in a communication network, we then have investigated multi-source localization techniques for underwater acoustics based on matched field (MF) processing (Chapter 4). We have developed a robust MF localization technique by solving a series of basis pursuit de-noising (BPDN) problems for multi-frequency signals. An efficient algorithm combining the homotopy approach and coordinate descent search has been developed for solving complex-valued BPDN problems. A position refinement method has been applied for further reduction of the complexity. We also have compared the proposed technique with other MF techniques using simulated data and real experimental data. The proposed technique outperforms other MF techniques in resolving sources positioned closely to each other, tolerance to the noise and capability of locating multiple sources.

Accurate knowledge of underwater environment is essential for MF localization techniques. In Chapter 5, we have investigated an MF inversion problem for estimating uncertain sound speed profile (SSP) in a deep water scenario, which requires solving a global optimization problem. Several global optimization algorithms based on simulated annealing have been compared and an adaptive hybrid algorithm combining a local optimization method (downhill simplex) with fast simulated annealing has been successfully applied for estimating the SSP.

Using the simulator developed in Chapter 3 to model underwater acoustic signal transmission, estimation of time-varying underwater acoustic channels has been investigated in Chapter 6. An iterative basis expansion model (BEM) based channel estimator using cubic B-splines has been developed. For dealing with channels of long delay spread, the complexity of the proposed estimator is relatively low, compared with the maximum likelihood and the minimum mean square error BEM based channel estimators. We have

shown that the proposed estimator has better estimation accuracy than the ML BEM estimator in dealing with time-varying multipath Rayleigh fading channels. We also have investigated the use of sparsity of underwater acoustic channels to improve the estimation accuracy. The non-zero channel taps are identified by solving a BPDN problem using the homotopy approach. It has been shown that by taking the sparsity into account, the channel estimation accuracy can be improved, especially in scenarios with low signal to noise ratios.

7.2 Future Work

Some suggestions for further work based on this thesis are given below:

1. We have verified the channel simulator using data from two real experiments. However, as mentioned in Chapter 3, these two experiments are of low frequency. It is useful to justify the simulator using higher frequency signals. Besides, channel time variations caused by the scattering from time-varying sea surface are not taken into account. It is useful to include the effect of surface variations into the simulator.

2. The developed MF localization technique has been proved to have better performance than several other MF techniques. It is useful to compare the proposed technique with other MF localization techniques. It is also useful to investigate the accuracy of localization with respect to the signal frequencies, size and number of elements of the receiver array as well as the number of sources. Data from other experiments with higher frequencies and larger number of sources is desirable.

3. In Chapter 6, sparse channel estimation of underwater acoustic channels is investigated using modeled signals. The communication system used in the simulation is relatively simple, compared with many other practical systems which have already been implemented in experiments. It is valuable to justify the proposed sparse channel estimation technique with more practical systems, involving such techniques as the error control coding and multi-carrier modulation.

Glossary

ASSA	**A**daptive **S**implex **S**imulated **A**nnealing
BEM	**B**asis **E**expansion **M**odel
BER	**B**it **E**rror **R**ate
BP	**B**asis **P**ursuit
BPDN	**B**asis **P**ursuit **D**e-**N**oising
CD	**C**oordinate **D**escent
CE	**C**omplex **E**xponentials
DHS	**D**own **H**ill **S**implex
DPS	**D**iscrete **P**rolate **S**pheroidal
FFP	**F**ast **F**ield **P**rogram
FSA	**F**ast **S**imulated **A**nnealing
KL	**K**arhunen-**L**oeve
MF	**M**atched **F**ield
MFI	**M**atched **F**ield **I**version
ML	**M**aximum **L**ikelihood
MMSE	**M**inimum **M**ean **S**quare **E**rror
MSE	**M**ean **S**quare **E**rror
MVDR	**M**inimum **V**ariance **D**istortionless **R**esponse
OFDM	**O**rthogonal **F**requency-**D**ivision **M**ultiplexing
IAA	**I**terative **A**daptive **A**lgorithm
MF	**M**atched **Field**
QPSK	**Q**uadrature **P**hase **S**hift **K**eying
SA	**S**imulated **A**nnealing
SNR	**S**ignal to **N**oise **R**atio
SSA	**S**implex **S**imulated **A**nnealing
SSP	**S**ound **S**peed **P**rofile

Bibliography

[1] M. B. Porter, "The KRAKEN normal mode program (Draft)," *Naval Research Laboratory, Washington DC*, 1992.

[2] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad Hoc Networks*, vol. 3, no. 3, pp. 257–279, 2005.

[3] I. F. Akyildiz, D. Pompili, and T. Melodia, "State-of-the-art in protocol research for underwater acoustic sensor networks," in *Proc. of the 1st ACM international workshop on Underwater networks*, 2006, pp. 7–16.

[4] J. Preisig, "Acoustic propagation considerations for underwater acoustic communications network development," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 4, pp. 2–10, 2007.

[5] J. Heidemann, W. Ye, J. Wills, A. Syed, and Y. Li, "Research challenges and applications for underwater sensor networking," in *Proc. of IEEE WCNC 2006, Las Vegas, USA*, 2006, vol. 1, pp. 228–235.

[6] L. M. Brekhovskikh and I. U. P. Lysanov, *Fundamentals of ocean acoustics*, Springer Verlag, 2003.

[7] M. Kocic, D. Brady, and M. Stojanovic, "Sparse equalization for real-time digital underwater acoustic communications," in *Proc. of OCEANS'95, San Diego, CA*, 1995, vol. 3, pp. 1417–1422.

[8] M. Stojanovic, "Recent advances in high-speed underwater acoustic communications," *IEEE Journal of Oceanic Engineering*, vol. 21, no. 2, pp. 125–136, 1996.

[9] S. J. Hwang and P. Schniter, "Efficient communication over highly spread underwater acoustic channels," in *Proceedings of the second workshop on Underwater networks*, 2007, pp. 11–18.

[10] W. Li and J. C. Preisig, "Estimation of rapidly time-varying sparse channels," *IEEE Journal of Oceanic Engineering*, vol. 32, no. 4, pp. 927–939, 2007.

[11] B. S. Sharif, J. Neasham, O. R. Hinton, and A. E. Adams, "A computationally efficient Doppler compensation system for underwater acoustic communications," *IEEE Journal of Oceanic Engineering*, vol. 25, no. 1, 2000.

[12] M. Stojanovic and J. Preisig, "Underwater acoustic communication channels: Propagation models and statistical characterization," *IEEE Communications Magazine*, vol. 47, no. 1, pp. 84–89, 2009.

[13] G. Bertolotto, T. Jenserud, and P. A. van Walree, "Initial design of an acoustic communication channel simulator," *UDT Europe 2007, Naples, Italy*, June 2007.

[14] A. B. Baggeroer, W. A. Kuperman, and P. N. Mikhalevsky, "An overiview of matched field methods in ocean acoustics," *IEEE Journal of Oceanic Engineering*, vol. 18, no. 4, pp. 401–424, 1993.

[15] H. P. Bucker, "Use of calculated sound fields and matched-field detection to locate sound sources in shallow water," *Journal of the Acoustical Society of America*, vol. 59, pp. 368–373, 1976.

[16] G. J. Orris, M. Nicholas, and J. S. Perkins, "The matched-phase coherent multi-frequency matched-field processor," *The Journal of the Acoustical Society of America*, vol. 107, pp. 2563–2575, 2000.

[17] C. Soares, M. Siderius, and S. M. Jesus, "Source localization in a time-varying ocean waveguide," *Journal of the Acoustical Society of America*, vol. 112, pp. 1879–1889, 2002.

[18] C. Soares and S. M. Jesus, "Broadband matched-field processing: Coherent and incoherent approaches," *Journal of the Acoustical Society of America*, vol. 113, pp. 2587–2598, 2003.

[19] M. Porter, R. Dicus, and R. Fizell, "Simulations of matched-field processing in a deep-water pacific environment," *IEEE Journal of Oceanic Engineering*, vol. 12, no. 1, pp. 173–181, 1987.

[20] T. Zemen and C. F. Mecklenbrauker, "Time-variant channel estimation using discrete prolate spheroidal sequences," *IEEE Transactions on Signal Processing*, vol. 53, no. 9, pp. 3597–3607, 2005.

[21] M. Visintin, "Karhunen-Loeve expansion of a fast Rayleigh fading process," *Electronics Letters*, vol. 32, no. 18, pp. 1712–1713, 1996.

[22] Z. Tang and G. Leus, "Time-multiplexed training for time-selective channels," *IEEE Signal Processing Letters*, vol. 14, no. 9, pp. 585–588, 2007.

[23] H. Mai, Y. V. Zakharov, and A. G. Burr, "Iterative channel estimation based on B-splines for fast flat fading channels," *IEEE Transactions on Wireless Communnications*, vol. 6, no. 4, pp. 1224–1229, Apr. 2007.

[24] M. B. Porter, "The time-matched fast-field program (FFP) for modeling acoustic pulse propagation," *Journal of the Acoustical Society of America*, vol. 87(5), pp. 2013–2023, May 1990.

[25] M. B. Porter and H. P. Bucker, "Gaussian beam tracing for computing ocean acoustic fields," *Journal of the Acoustical Society of America*, vol. 82(4), pp. 1349–1359, Oct. 1987.

[26] V. Červenỳ, M.M. Popov, and I. Pšenčík, "Computation of wave fields in inhomogeneous mediagaussian beam approach," *Geophysical Journal of the Royal Astronomical Society*, vol. 70, no. 1, pp. 109–128, 1982.

[27] W. H. Munk, "Sound channel in an exponenntially stratified ocean, with application of SOFAR," *Journal of the Acoustical Society of America*, vol. 65(5), pp. 220–226, February 1974.

[28] C. Bjerrum-Niese and R. Lutzen, "Stochastic simulation of acoustic communication in turbulent shallow water," *IEEE Journal of Oceanic Engineering*, vol. 25, no. 4, pp. 523–532, 2000.

[29] T. H. Eggen, A. B. Baggeroer, and J. C. Preisig, "Communication over Doppler spread channels. Part I: Channel and receiver presentation," *IEEE Journal of Oceanic Engineering*, vol. 25, no. 1, pp. 62–71, 2000.

[30] M. Patzold, U. Killat, F. Laue, and Y. Li, "On the statistical properties of deterministic simulation models for mobile fading channels," *IEEE Transactions on Vehicular Technology*, vol. 47, no. 1, pp. 254–269, 1998.

[31] Y. R. Zheng and C. Xiao, "Simulation models with correct statistical properties for rayleigh fading channels," *IEEE Transactions on Communications*, vol. 51, no. 6, pp. 920–928, 2003.

[32] C. Xiao, Y. R. Zheng, and N. C. Beaulieu, "Novel sum-of-sinusoids simulation models for rayleigh and rician fading channels," *IEEE Transactions on Wireless Communications*, vol. 5, no. 12, pp. 3667–3679, 2006.

[33] X. Geng and A. Zielinski, "An eigenpath underwater acoustic communication channel model," in *Proc. of Oceans '95, San Diego, CA, USA*, 1995, vol. 2, pp. 1188–1196.

[34] G. L. Stüber, *Principles of mobile communication*, Springer Netherlands, 2001.

[35] B. Li, S. Zhou, M. Stojanovic, L. Freitag, and P. Willett, "Multicarrier communication over underwater acoustic channels with nonuniform Doppler shifts," *IEEE Journal of Oceanic Engineering*, vol. 33, no. 2, pp. 198–209, 2008.

[36] S. Mason, C. Berger, S. Zhou, and P. Willett, "Detection, synchronization, and Doppler scale estimation with multicarrier waveforms in underwater acoustic communication," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 9, pp. 1638–1649, 2008.

[37] C. R. Berger, S. Zhou, J. C. Preisig, and P. Willett, "Sparse channel estimation for multicarrier underwater acoustic communication: From subspace methods to compressed sensing," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1708–1721, 2010.

[38] S.M. Flatte, "Wave propagation through random media: Contributions from ocean acoustics," *Proceedings of the IEEE*, vol. 71, no. 11, pp. 1267–1294, 1983.

[39] C. Bjerrum-Niese, L. Bjorno, M. A. Pinto, and B. Quellec, "A simulation tool for high data-rate acoustic communication in a shallow-water, time-varying channel," *IEEE Journal of Oceanic Engineering*, vol. 21, no. 2, pp. 143–149, 1996.

[40] M. Siderius and M. B. Porter, "Modeling broadband ocean acoustic transmissions with time-varying sea surfaces," *Journal of the Acoustical Society of America*, vol. 124, pp. 137, 2008.

[41] D. B. Kilfoyle and A. B. Baggeroer, "The state of the art in underwater acoustic telemetry," *IEEE Journal of Oceanic Engineering*, vol. 25, no. 1, pp. 4–27, 2000.

[42] A. Essebbar, G. Loubet, and F. Vial, "Underwater acoustic channel simulations for communication," in *Proc. of OCEANS'94, Brest, France*, 1994, vol. 3, pp. 495–500.

[43] S. Yerramalli and U. Mitra, "On optimal resampling for ofdm signaling in doubly-selective underwater acoustic channels," in *Proc. of OCEANS 2008, Quebec, Canada*, 2008, pp. 1–6.

[44] Y. V. Zakharov and V. P. Kodanev, "Multipath-Doppler diversity of OFDM signals in an underwater acoustic channel," in *Proc. of IEEE ICASSP 2000, Istanbul, Turkey*, 2000, vol. 5, pp. 2941–2944.

[45] L. Yu and L. B. White, "Optimum receiver design for broadband doppler compensation in multipath/doppler channels with rational orthogonal wavelet signaling," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4091–4103, 2007.

[46] Y. V. Zakharov and V. P. Kodanev, "Doppler scattering adapted reception in a hydroacoustic communication channel," *Acoustocal Physics*, vol. 41, no. 2, pp. 219–223, 1995.

[47] G. W. Wornell, "A Karhunen-Loeve-like expansion for 1/f processes via wavelets," *IEEE Transactions on Information Theory*, vol. 36, no. 4, pp. 859–861, 1990.

[48] M. K. Tsatsanis and G. B. Giannakis, "Modelling and equalization of rapidly fading channels," *Int. J. Adaptive Contr. Signal Processing*, vol. 10, no. 2-3, pp. 159–176, 1996.

[49] F. Qu and L. Yang, "Basis expansion model for underwater acoustic channels?," in *Proc. of OCEANS 2008, Quebec, Canada*, 2008, pp. 1–7.

[50] Y. V. Zakharov, T. C. Tozer, and J. F. Adlard, "Polynomial spline-approximation of Clarke's model," *IEEE Transactions on Signal Processing*, vol. 52, no. 5, pp. 1198–1208, 2004.

[51] N. P. Korneichuk, *Exact constants in approximation theory*, Cambridge University Press, 1991.

[52] V. M. Tikhomirov, "Harmonics and splines as optimal tools for approximation and recovery," *Russian Mathematical Surveys*, vol. 50, pp. 355–402, 1995.

[53] M. Unser, A. Aldroubi, and M. Eden, "B-Spline Signal Processing: Part I Theory," *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 821–833, 1993.

[54] V. A. Zheludev, "Asymptotic formulae for local spline-approximation on a uniform grid," *Doclady Academii Nauk*, vol. 269, no. 4, pp. 797–802, 1983.

[55] V. A. Zheludev, "Local spline-approximation on a uniform grid," *Computational Mathematics and Mathematical Physics*, vol. 27, no. 9, pp. 1296–1310, 1987.

[56] Y. V. Zakharov and T. C. Tozer, "Frequency estimator with dichotomous search of periodogram peak," *Electronics Letters*, vol. 35, no. 19, pp. 1608–1609, Sept. 1999.

[57] S. D. Chuprov, "Interference structure of an acoustic field in a layered waveguide," *Acoustics of the Ocean*, pp. 71–91, 1982.

[58] G. A. Grachev, "Theory of acoustic field invariants in layered waveguides," *Acoustical Physics*, vol. 39, no. 1, pp. 33–35, 1993.

[59] M. Porter, "Bellhop gaussian beam/finite element beam code," Available in the Acoustics Toolbox, `http://oalib.hlsresearch.com/Rays`.

[60] J. Murray and D. Ensberg, "The SWellEx-96 Experiment," `http://www.mpl.ucsd.edu/swellex96/`.

[61] T. Chen, C. Liu, and Y. V. Zakharov, "Matched-phase coherent broadband matched-field processor using phase descent search," in *Proc. of the 10th European Conference on Underwater Acoustics (ECUA10), Istanbul, Turkey*, 2010, pp. 590–595.

[62] C. Liu, T. Chen, and Y. V. Zakharov, "Source localization using sparsity based iterative adaptive beamforming," in *Proc. of the 10th European Conference on Underwater Acoustics (ECUA10), Istanbul, Turkey*, 2010, pp. 604–610.

[63] Z. Xu, Y. V. Zakharov, and V. P. Kodanev, "Space-time signal processing of OFDM signals in fast-varying underwater acoustic channel," in *Proc. of OCEANS'2007-Europe, Aberdeen, Scotland*, 2007, pp. 1–6.

[64] J. G. Proakis, *Digital communications*, McGraw-Hill, New York, 1995.

[65] Y. V. Zakharov, V. M. Baronkin, and T. C. Tozer, "DFT-based frequency estimators with narrow acquisition range," *IEE Proceedings Communications*, vol. 148, no. 1, pp. 1–7, 2001.

[66] R. H. Owen, B. V. Smith, and R. F. W. Coates, "An experimental study of rough surface scattering and its effects on communication coherence," in *Proc. of OCEANS'94, Brest, France*, 1994, vol. 3, pp. 483–488.

[67] Y. V. Zakharov and V. P. Kodanev, "Noise immunity of adaptive reception of complex acoustic signals in the presence of reflections from ocean boundaries," *Acoustocal Physics*, vol. 42, no. 2, pp. 185–191, 1996.

[68] M. Badiey, Y. Mu, J. A. Simmen, and S. E. Forsythe, "Signal variability in shallow-water sound channels," *IEEE Journal of Oceanic Engineering*, vol. 25, no. 4, pp. 492–500, 2000.

[69] W. Xu, A. B. Baggeroer, and H. Schmidt, "Performance analysis for matched-field source localization: Simulations and experimental results," *IEEE Journal of Oceanic Engineering*, vol. 31, no. 2, pp. 325–344, 2006.

[70] N. O. Booth, P. A. Baxley, J. A. Rice, P. W. Schey, W. S. Hodgkiss, G. L. D'Spain, and J. J. Murray, "Source localization with broad-band matched-field processing in shallow water," *IEEE Journal of Oceanic Engineering*, vol. 21, no. 4, pp. 402–412, 1996.

[71] T. Yardibi, J. Li, P. Stoica, M. Xue, and A. B. Baggeroer, "Source localization and sensing: A nonparametric iterative adaptive approach based on weighted least squares," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 1, pp. 425–443, 2010.

[72] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations," *Journal of Constructive approximation*, vol. 13, no. 1, pp. 57–98, 1997.

[73] D. L. Donoho and Y. Tsaig, "Fast solution of l1-norm minimization problems when the solution may be sparse ," *IEEE Transactions on Information Theory*, vol. 54, no. 11, pp. 4789–4812, 2008.

[74] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution," *Communications on pure and applied mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

[75] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.

[76] E. Van Den Berg and M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," *SIAM Journal on Scientific Computing*, vol. 31, no. 2, pp. 890–912, 2008.

[77] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra and its Applications*, vol. 284, no. 1-3, pp. 193–228, 1998.

[78] E. Zamanizadeh, J. Gomes, and J. Bioucas-Dias, "Identification of sparse time-varying underwater channels through basis pursuit methods," in *Proc. of the 10th European Conference on Underwater Acoustics (ECUA10), Istanbul, Turkey*, 2010, pp. 1054–1061.

[79] D. Malioutov, M. Cetin, and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 3010–3022, 2005.

[80] D.M. Malioutov, M. Cetin, and A.S. Willsky, "Homotopy continuation for sparse signal representation," in *Proc. of IEEE ICASSP 2005, Philadelphia, PA, USA*, 2005, vol. 5, pp. v–733.

[81] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, no. 1, pp. 1–22, Feb. 2010.

[82] Y. V. Zakharov and T. C. Tozer, "Multiplication-free iterative algorithm for LS problem," *Electronics letters*, vol. 40, no. 9, pp. 567–568, Apr. 2004.

[83] D. M. Malioutov, M. Cetin, and A. S. Willsky, "Optimal sparse representations in general overcomplete bases," in *Proc. of IEEE ICASSP 2004, Montreal, Quebec, Canada*, 2004, vol. 2, pp. 793–796.

[84] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization," *Proc. Nat. Acad. of Sci.*, vol. 100, no. 5, pp. 2197–2202, 2003.

[85] Z. Yang, R.C. de Lamare, and X. Li, "L1 regularized stap algorithm with a generalized sidelobe canceler architecture for airborne radar," *IEEE Transactions on Signal Processing*, vol. 60, pp. 674–686, 2012.

[86] M.B. Porter, "The KRAKEN normal mode program," *Navel Research Laboratory, Washington DC*, 1992.

[87] H. Schmidt, A. B. Baggeroer, W. A. Kuperman, and E. K. Scheer, "Environmentally tolerant beamforming for high-resolution matched field processing: Deterministic mismatch," *The Journal of the Acoustical Society of America*, vol. 88, pp. 1851, 1990.

[88] S. E. Dosso, M. J. Wilmut, and A. S. Lapinski, "An adaptive-hybrid algorithm for geoacoustic inversion," *IEEE Journal of Oceanic Engineering*, vol. 26(3), pp. 324–336, July 2001.

[89] D. F. Gingras and P. Gerstoft, "Inversion for geometric and geoacoustic parameters in shallow water: Experimental results," *Journal of the Acoustical Society of America*, vol. 97(6), pp. 3589–3598, June 1995.

[90] M. D. Collins and W. A. Kuperman, "Focalization: environmental focusing and source localization," *Journal of the Acoustical Society of America*, vol. 90, pp. 1410–1422, Sept. 1991.

[91] M. T. Fallat and S. E. Dosso, "Geoacoustic inversion via local, global and hybrid algorithms," *Journal of the Acoustical Society of America*, vol. 105(6), pp. 3219–3230, June 1999.

[92] C. E. Lindsay and N. R. Chapman, "Matched field inversion for geoacoustic model parameters using adaptive simulated annealing," *IEEE Journal of Oceanic Engineering*, vol. 18, pp. 224–231, July 1993.

[93] P. Gertoft, "Inversion of seismoacoustic data using genetic algorithms and a posteriori probability distributions," *Journal of the Acoustical Society of America*, vol. 95(2), pp. 770–782, 1984.

[94] M.J. Wilmut D. Tollefsen and R.Chapman, "Estimates of geoacoustic model parameters from inversions of horizontal and vertical line array data," *IEEE Journal of Oceanic Engineering*, vol. 30(4), pp. 764–772, 2005.

[95] C.Huang and W.S. Hodgkiss, "Matched-field geozoustic inversion of low-frequency source tow data from the ASIAEX East China Sea experiment," *IEEE Journal of Oceanic Engineering*, vol. 29(4), pp. 952–963, 2004.

[96] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671, 1983.

[97] H. Szu and R. Hartley, "Nonconvex optimization by fast simulated annealing," *Proceedings of the IEEE*, vol. 75(11), pp. 1538–1540, November 1987.

[98] H. Szu and R. Hartley, "Fast simulated annealing," *Physics Letters A*, vol. 122, pp. 157–162, June 1987.

[99] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, pp. 308–313, 1965.

[100] M. B. Porter and E. L. Reiss, "A numerical method for bottom interacting ocean acoutic normal modes," *Journal of the Acoustical Society of America*, vol. 77(5), pp. 1760–1767, 1985.

[101] M. Stojanovic, L. Freitag, and M. Johnson, "Channel-estimation-based adaptive equalization of underwater acoustic signals," in *Proc. of OCEANS'99, Seattle, USA*, 1999, vol. 2, pp. 590–595.

[102] M. Stojanovic, J. A. Catipovic, and J. G. Proakis, "Phase-coherent digital communications for underwater acoustic channels," *IEEE Journal of Oceanic Engineering*, vol. 19, pp. 100–111, 1994.

[103] H. V. Poor and G. W. Wornell, *Wireless Communications: Signal Processing Perspectives*, Prentice Hall Press, 1998.

[104] Y. V. Zakharov, T. C. Tozer, and J. F. Adlard, "Polynomial spline-approximation of Clarke's model," *IEEE Transactions on Signal Processing*, vol. 52, no. 5, pp. 1198–1208, 2004.

[105] Y. V. Zakharov and T. C. Tozer, "Local splines in signal processing applications: low complexity, high accuracy," in *Proc. Inter. Workshop on Signal Processing for Wireless Commun.(SPWC04)*.

[106] Y. V. Zakharov, V. M. Baronkin, and J. Zhang, "Optimal detection of qam signals in fast fading channels with imperfect channel estimation," in *Proc. of ICASSP 2008, Las Vegas, Nevada, USA*, 2008, pp. 3205–3208.

[107] S. M. Kay, *Fundamentals of statistical signal processing: estimation theory*, Prentice-Hall, 1993.

[108] R. N. Khal, Y. V. Zakharov, and J. Zhang, "B-spline based joint channel and frequency offset estimation in doubly-selective fading channels," in *Proc. of IEEE ICASSP 2010, Dallas, Texas, USA*, 2010, pp. 3214–3217.

[109] Y. V. Zakharov, V. M. Baronkin, and J. Zhang, "Optimal and mismatched detection of qam signals in fast fading channels with imperfect channel estimation," *IEEE Transactions on Wireless Communications*, vol. 8, no. 2, pp. 617–621, 2009.

[110] R. Galvin and R. F. W. Coates, "Analysis of the performance of an underwater acoustic communications system and comparison with a stochastic model," in *Proc. of OCEANS'94, Brest, France*, 1994, vol. 3, pp. 478–482.

[111] A. Essebbar and V. Vercelloni, "Simulation of communication system for underwater acoustics," in *Proc. of OCEANS'95, San Diego, CA, USA*, 1995, pp. 1204–1207.

[112] S. F. Cotter and B. D. Rao, "Sparse channel estimation via matching pursuit with application to equalization," *IEEE Transactions on Communications*, vol. 50, no. 3, pp. 374–377, 2002.

[113] S. Ozen, W. Hillery, M. Zoltowski, S. M. Nereyanuru, and M. Fimoff, "Structured channel estimation based decision feedback equalizers for sparse multipath channels with applications to digital tv receivers," in *Proc. of Thirty-Sixth Asilomar*

Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 2002, vol. 1, pp. 558–564.

[114] C. Liu, T. Chen, and Y. V. Zakharov, "Broadband underwater source localization by solving basis pursuit de-noising using coordinate descent search," in *Proc. of IEEE 7th ISWCS, York, UK*, 2010, pp. 1–5.